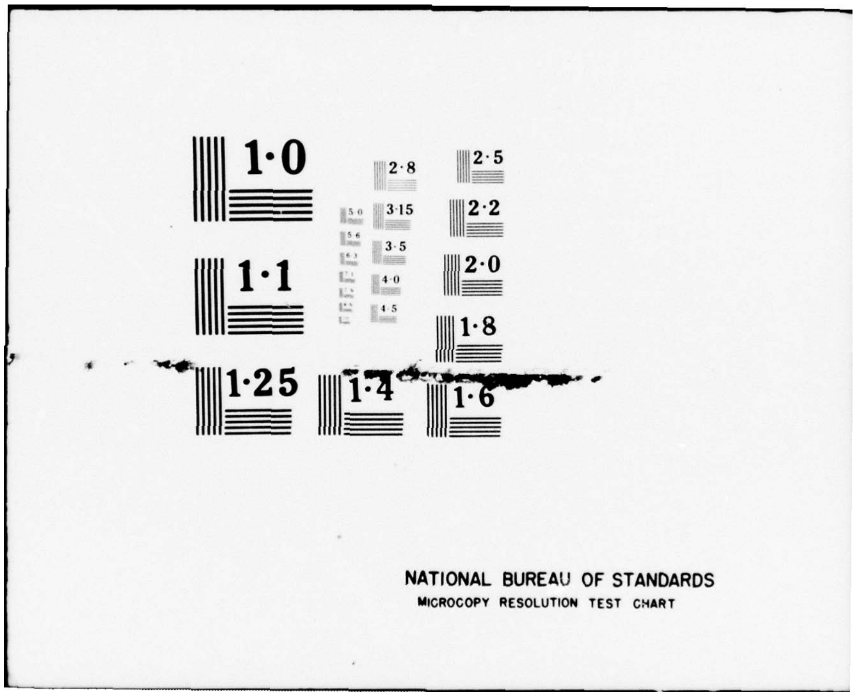
AD-A040 832 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 5/9

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS) DOCUMENTATION.(U)

MAY 77 J D THOMAS, G E ARMSTRONG

UNCLASSIFIED CAA-D-77-1 NL

1 OF 5
ADA
040832


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

DOCUMENTATION
CAA - D - 77 - 1
AD

3

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS)

DOCUMENTATION

MAY 1977

PREPARED BY

METHODOLOGY, RESOURCES AND COMPUTATION DIRECTORATE

US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20014

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CAA-D-77-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Officer Dual Specialty Allocation System (ODSAS) Documentation.		5. TYPE OF REPORT & PERIOD COVERED Model Documentation Rept. for January 1976-May 1977,
		6. PERFORMING ORG. REPORT NUMBER CAA-D-77-1
7. AUTHOR(s) MAJ Joseph D. Thomas Mr. George E. Armstrong		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office, Deputy Chief of Staff For Personnel Headquarters, Department of the Army		12. REPORT DATE May 1977
		13. NUMBER OF PAGES 468 (12) 447p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) US Army Military Personnel Center 200 Stovall Street Alexandria, VA 22331		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
RECEIVED
JUN 6 1977
A

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Prepared in cooperation with the Officer Personnel Management Directorate, US Army Military Personnel Center, 200 Stovall Street, Alexandria, VA 22331

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Army Personnel, Officer Personnel Management System, Dual Specialty Allocation, Dual Qualified Officers, Basic Entry Specialty, Alternate Specialty, Personnel Management, Automated Information System, Model Documentation, Linear Programing, Officer Dual Specialty Allocation System Methodology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In 1974 the Army adopted the Officer Personnel Management System (OPMS), replacing the earlier "career branch" system under which the officer corps had been managed. Under OPMS each officer is assigned two specialties--a basic entry specialty (BES), and an alternate. Goals of OPMS include improving the match between the officer's qualifications and the Army's requirements and providing discrete career development patterns for the individual officer in both his BES and alternate specialty. The US Army Concepts Analysis
(cont on p 1473B)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390996

(cont'd fr p 1473A) This agency (CAA) has developed ODSAS,

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Agency (CAA), at the request of the Office of the Deputy Chief of Staff for Personnel (ODCSPER), Headquarters Department of the Army, developed the Officer Dual Specialty Allocation System (ODSAS), a computer-based system, to assist OPMS managers in satisfying Army officer personnel requirements. The ODSAS-derived solution is driven by requirements associated with any force structure specified by the user. The methodology ~~employed~~ computes the optimum number of officers for allocation to specific OPMS specialty pairings, consistent with the specified force structure requirements. The system treats officer grades from lieutenant through colonel, inclusively, over a period of time (up to ~~nine~~ 9 years). This documentation comprises the users' and technical manual for operating the automated information system by which the methodology is implemented and results displayed.

report

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

SYSTEM DOCUMENTATION

May 1977

Prepared by

Methodology, Resources and Computation Directorate

US Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, Maryland 20014

FOREWORD

DEPARTMENT OF THE ARMY
US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20014

IN REPLY REFER TO

MOCA-MRM 20 May 1977

SUBJECT: System Documentation of the Officer Dual Specialty Allocation System (ODSAS)

Deputy Chief of Staff for Personnel
Department of the Army
Washington, DC 20310

1. References:

AD A037 456

a. Study report, CAA-SR-76-6, "Officer Dual Specialty Allocation System (ODSAS)," US Army Concepts Analysis Agency, April 1976.

b. Report, "Officer Dual Specialty Allocation System (ODSAS), Management Overview," US Army Concepts Analysis Agency, September 1976.

2. Reference a contains a detailed description of the ODSAS methodology and the computer-based information system implementing the methodology. Reference b highlights the salient features of both, and comprises an introduction to the ODSAS for personnel managers in ODCSPER and MILPERCEN.

3. The inclosed system documentation comprises the users' and technical manual, and is designed for use by analysts and computer programers in applying and maintaining the ODSAS.

1 Incl
as

Ennis C Whitehead
ENNIS C. WHITEHEAD
Major General, USA
Commanding

TABLE OF CONTENTS

	Page
Disclaimer.	ii
Report Documentation Page, DD Form 1473	iii
Title Page.	v
Foreword.	vii
Letter of Transmittal.	ix
Table of Contents	xi
List of Figures	xiv
List of Tables.	xvii

CHAPTER

I.	Introduction.	I-1
	Overview of the ODSAS	I-1
	Scope of Documentation.	I-2
II.	Preparation for ODSAS Processing.	II-1
	Purpose	II-1
	Types of Input.	II-1
	User Decisions.	II-4
III.	Automated Information System Design	III-1
	Purpose	III-1
	System Design	III-1
	Discussion and Flow Charts for Initialization Phase	III-12
	Discussion and Flow Charts for Matrix Generator Activity.	III-16
	Discussion and Flow Chart for FMPS Solution Activity.	III-28
	Discussion and Flow Charts for Data Base Creation Activity	III-30
	Discussion and Flow Chart for On-Line Inquiry Activity.	III-34
	Discussion and Flow Chart for Segment Linkages Activity.	III-36
	Separating Active and Inactive Records in the Cumulative Data Base.	III-37
	Special Procedure for Accessing All Records in the Cumulative Data Base.	III-38

TABLE OF CONTENTS

CHAPTER		Page
IV.	Input Cards Formats	IV-1
	Purpose	IV-1
	Card Inputs for Initialization Phase.	IV-1
	Card Inputs for Processing Phase.	IV-8
V.	ODSAS Printed Output.	V-1
	Purpose	V-1
	Initialization Phase Reports.	V-1
	Processing Phase Reports.	V-5
VI.	Operations Guide.	VI-1
	Purpose	VI-1
	Catalogued Runstreams	VI-1
	Operating Instructions for Initialization Phase	VI-1
	Operating Instructions for Processing Phase . . .	VI-5
	Procedures to Ascertain the State of Processing of the System Segments.	VI-9
	Procedures to Recover from Premature Termination of System Segment	VI-11
	Procedures for Modifying the LP Problem for Rerunning	VI-12
	Procedures for Saving the MIRADS Data Bases . . .	VI-13
	General Procedures for On-Line Inquiry of the Data Bases.	VI-14
	Standard Query Sets for Data Base Access.	VI-15
	Modifications to Standard Query Sets.	VI-15
VII.	Data Base Record Descriptions	VII-1
	Purpose	VII-1
	Types of Records.	VII-1
	Data Base Record Formats.	VII-1
VIII.	Computer Disc and Tape File Descriptions.	VIII-1
	Purpose	VIII-1
	Disc and Tape File Naming Convention.	VIII-1
	Index of ODSAS Disc and Tape File Descriptions. .	VIII-1

TABLE OF CONTENTS

		Page
CHAPTER		
IX.	Catalogued Runstreams	IX-1
	Purpose	IX-1
	Initialization Phase Runstreams	IX-4
	Processing Phase Runstreams	IX-8
	Ancillary Runstreams.	IX-29
X.	Dictionary of Principal Variables	X-1
	Purpose	X-1
	Data Dictionary for the Initialization Phase. . .	X-1
	Data Dictionary for the Processing Phase.	X-8
XI.	Essential Program Narratives.	XI-1
	Purpose	XI-1
	Initialization Phase Program Narratives	XI-2
	Processing Phase Program Narratives	XI-8
XII.	Program Listings.	XII-1
	Purpose	XII-1
	Index of Source Program Listings.	XII-1
APPENDIXES		
A.	Documentation Contributors.	A-1
B.	References.	B-1
C.	Glossary.	C-1
D.	ODSAS Naming Conventions.	D-1
E.	Distribution.	E-1

LIST OF FIGURES

Figure		Page
III-1	ODSAS Automated Information System, Initialization Phase	III-3
III-2	ODSAS Automated Information System, Processing Phase	III-6
III-3	Matrix Generator, Processing Phase Major Activity Number 1.	III-7
III-4	FMPS, Processing Phase Major Activity Number 2	III-8
III-5	Data Base Creation, Processing Phase Major Activity Number 3.	III-9
III-6	On-Line Inquiry, Processing Phase Major Activity Number 4.	III-11
III-7	Linkage, Processing Phase Major Activity Number 5	III-12
III-8	Structure of Runstream PFCAA.SACS	III-13
III-9	Structure of Runstream PFCAA.TOUR	III-15
III-10	Structure of Runstream PFCAA.INITIAL	III-16
III-11	Matrix Generator Flow Chart with Internal File Names Specified	III-17
III-12	Flow Chart of Matrix Generator Activity for the Colonel Segment.	III-19
III-13	Flow Chart of Matrix Generator Activity for the Lieutenant Colonel and Major Segments. . .	III-21
III-14	Flow Chart of Matrix Generator Activity for the Captain Segment.	III-23
III-15	Flow Chart of Matrix Generator Activity for the Lieutenant Segment	III-25
III-16	General Sequence of FMPS Control Language Procedures	III-29
III-17	Flow Chart for Calling Sequence of Programs Used in Data Base Creation Activity.	III-30
III-18	Interaction of the Data Base Creation Programs with the Input and Output Files Needed to Create the Data Bases.	III-31
III-19	Interaction of Data Base Creation Program with the Input and Output Files Needed to Re-sequence and Update Cumulative Data Base Input File	III-33
III-20	Flow Chart for Loading the Data Bases in the Data Base Creation Activity.	III-34
III-21	Flow Chart of the UPDATE Program Procedures. . .	III-35
III-22	Flow Chart of the Linkage Activity Procedures. .	III-36
III-23	Flow Chart of Procedure for Separating Active and Inactive Records in the Cumulative Data Base	III-38

LIST OF FIGURES

Figure		Page
IV-1	Examples of Utilization Ratio and Tour Lengths Input Data Cards (File ODTURUDØ1). . .	IV-3
IV-2	Examples of Combined Input Parameters/Rates and Populations Data Cards (File ODR8SUDØ1). .	IV-9
IV-3	Examples of Cards to Update Specialty Requirements File (ODSACUDØ1) and/or Attrition and Promotion Rates File (ODPOPUDØ1).	IV-15
IV-4	Examples of Cards to Modify Data Values in an Existing FMPS Solution	IV-17
V-1	Sample PERSACS Requirements Report - by Specialty (for Specialty 49)	V-2
V-2	Sample PERSACS Requirements Report - by Grade (for Grade 6).	V-3
V-3	Excerpt of Attrition and Promotion Rates Computation Report	V-4
V-4	ODSAS Statistical Summary Report--Part 1	V-6
V-5	ODSAS Statistical Summary Report--Part 2	V-7
V-6	Sample of Matrix Generator Activity Optional Report, Part 1--Restrictive Flow Codes for Grade n at Year y.	V-8
V-7	Example of Redundant Flow Control Constraint . .	V-9
V-8	Sample of Matrix Generator Activity Optional Report, Part 2--Restrictive Flow Codes for the Higher (Promotee) Grade.	V-11
V-9	Sample of Matrix Generator Activity Optional Report, Part 3--Last Exit Year Points for Grade n.	V-13
V-10	Sample Matrix Statistics Output from FMPS. . . .	V-14
V-11	FMPS Solution Output--Identifier Section	V-14
V-12	FMPS Solution Output--Rows Section	V-16
V-13	FMPS Solution Output--Columns Section.	V-17
V-14	Excerpt of Direct Access Information Retrieval Display for Number of Officers by Grade in Each Specialty Pair.	V-20
V-15	Excerpt of Direct Access Information Retrieval Display of Specialty Designations for Captains	V-21
V-16	Excerpt of Direct Access Information Retrieval Display for Number of Officers by Grade and Specialty Pairs Assigned to Specialty 11 . . .	V-22
V-17	Excerpt of Direct Access Information Retrieval Display of Grade Substitution by Specialty Within Grade and Year.	V-23

LIST OF FIGURES

Figure		Page
V-18	Excerpt of Unfilled Requirements Report	V-24
V-19	Example of ODSAS Procurement Report, Part One (BES Requirements, Plus AES Requirements Prorated to All Specialties).	V-26
V-20	Example of ODSAS Procurement Report, Part Two (BES Requirements, Plus AES Requirements Prorated to Non-combat Arms Specialties Only.	V-28
V-21	Example of ODSAS Procurement Report, Option 1 (Part Two, Modified by Proration of Requirements for Specified AES to Both Combat and Non-combat Arms Specialties).	V-29
XI-1	Example of Arcs Identifiable as Containing Flow and Thus Assigned a Code Number by the RESLO Program (for Grades 6, 5 and 4) . . .	XI-14
XI-2	Illustrative Example of the General Procedure in LODIAG Program for Defining X Arc Variables and Coefficients for FMPS COLUMN Chapter Input	XI-25
XI-3	Illustrative Example of the LODIAG Program Procedure for Defining X and Y Arc Variables and Coefficients when Promotion Considerations are Included (Grades 5 and 4), for FMPS COLUMN Chapter Input	XI-26
XI-4	Illustrative Example of the CPDIAG Program Procedure for Defining Y Arc Constraint Names and Coefficients (Grade 3) for FMPS COLUMN Chapter Input.	XI-30

LIST OF TABLES

Table		Page
II-1	Data Input Responsibilities	II-2
III-1	Identification of Minor Supporting Functions or Subroutines Used in Matrix Generator Activity Flow Charts.	III-27
V-1	Summary of Application of Variables, Constraints and Promotions in the ODSAS Grade Segments. . .	V-18
VI-1	EXEC 8 Commands for Processing Phase of ODSAS . .	VI-6
VI-2	Standard Query Sets for Data Base Access.	VI-17
VII-1	CURSEG Row Record Format.	VII-3
VII-2	CURSEG Column Record Format	VII-4
VII-3	CUMSEG Row Record Format.	VII-5
VII-4	CUMSEG Column Record Format	VII-6
XI-1	Format of Valid Output Records Produced by the SACSEXTRACT Program.	XI-3
XI-2	Storage in Array IRAY(1-10) of Records Decoded by the SACSPREPRO Program	XI-3
XI-3	Conversion of Certain Nonstandard Specialty Numbers to Authorized OPMS Specialty Numbers by the SACSPREPRO Program	XI-4
XI-4	Redistribution of Other Non-OPMS Specialty Requirements into Authorized OPMS Specialties by the SACSCREATE Program	XI-6
XI-5	Parameters Edited by the INITIAL Program.	XI-8
XI-6	Criteria Governing Selection, by the ROWCHP Program, of Year-Dependent Constraint Types and Names for Grades 6, 5 and 4	XI-10
XI-7	Additional Constraint Types and Names Produced by the ROWCHP Program for Grades 6, 5 and 4 . .	XI-11
XI-8	Criteria Governing Selection, by the ROWCHP Program, of Year-Dependent Constraint Types and Names for Grade 3	XI-12
XI-9	Criteria Governing Selection, by the ROWCHP Program, of Year-Dependent Constraint Types and Names for Grade 2	XI-13
XI-10	Code Numbers Generated by the RESLO Program to Indicate When Promotees in Grades 5 and 4 Transit from One Specialty to Another	XI-15
XI-11	Variable Names and Coefficients (for Grades 6, 5 and 4) Defined by the LOCOLS Program for FMPS COLUMN Chapter Input	XI-18
XI-12	Variable Names and Coefficients (for Grade 3) Defined by the LOCOLC Program for FMPS COLUMN Chapter Input.	XI-20

LIST OF TABLES

Table		Page
XI-13	Variable Names and Coefficients (for Grade 2) Defined by the LOCOLL Program for FMPS COLUMN Chapter Input.	XI-23
XI-14	XNn Variable Names and Coefficients Defined by the LODIAG Program for FMPS COLUMN Chapter Input	XI-28
XI-15	Variable Names and Coefficients Defined by the HICOLS Program for FMPS COLUMN Chapter Input. .	XI-29
XI-16	Constraint Names and Coefficients (for Grade 3) Defined by the CPDIAG Program for FMPS COLUMN Chapter Input.	XI-31
XI-17	Nonzero b-coefficients Defined by the RHS Program for FMPS RIGHT HAND SIDE Chapter Input	XI-31
XI-18	Ranges on Selected b-coefficients Computed by the RANGE Program for FMPS RANGE Chapter Input	XI-32
XI-19	Upper, Lower, and Fixed Bounds Defined by the BOUNDS Program for FMPS BOUNDS Chapter Input. .	XI-32
XI-20	Output Record Format for the NEWSAV Program . . .	XI-35
XI-21	Output Record Format for the COMBIN Program . . .	XI-36
XI-22	Format of Records Written to Temporary Cumulative Data Base Input File ODSAPUDØ3 by the RECORD Program	XI-39
XI-23	Updating Actions Taken by the LINKAGE Program . .	XI-44
D-1	Correlation of Naming Convention Formats and Constraint Types.	D-6

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER I
INTRODUCTION

1. Overview of the ODSAS

a. The ODSAS employs a linear programing-based methodology to determine the optimum allocation of OPMD-managed* officers to pairs of Officer Personnel Management System (OPMS) specialties. The ODSAS automated information system was designed in the second phase of a two phase study sponsored by the Office of the Deputy Chief of Staff for Personnel (ODCSPER), HQ DA, and conducted by the US Army Concepts Analysis Agency (CAA). The first phase of the study entailed determining if a feasible methodology could be formulated. In August 1975, a feasible methodology was established; this methodology was incorporated in a computer-based information system.

b. The ODSAS automated information system is composed of a series of large linear programing (LP) problems. Each of these LP problems is solved sequentially by grade, starting with a LP problem for the allocation of colonels to the OPMS specialty pairs. During the processing, data on the LP solutions are available for on-line inquiry by the user. Overall, the information system is designed to maximize the computer's contribution to the processing, control, and solution and analysis of the LP problem, but leaving interpretation of results to the user.

c. The system implementation effort was accomplished in two phases. First, the information system was designed, developed, and tested with sample data on the UNIVAC 11Ø8 computer located at CAA. Second, all the programs and necessary runstreams were loaded onto the UNIVAC 11Ø8 at MILPERCEN (System 2 was designated to accommodate the ODSAS).** Subsequently, the input files with real data were created and operational testing was conducted at MILPERCEN.

---

*OPMD - Officer Personnel Management Directorate

**The symbol "Ø" is used to represent the digit zero whenever that digit could be confused with the capital letter "O." Except in reproductions of computer printouts this convention applies throughout the remainder of this documentation.

2. Scope of Documentation

a. This documentation supplements the ODSAS Study Report (CAA SR-76-6, dated April 1976). Access by the reader to that report and the references cited therein is assumed. Topics fully discussed in the study report are treated only briefly in this documentation.

b. This publication is designed as a user's manual for both analysts and computer programers. A minimum working level knowledge of the UNIVAC Exec 8 control language on the part of all users is also assumed.

c. The content of this documentation is organized into chapters by subject-matter areas, with the content level of the respective chapters based on anticipated needs of the analyst or the programer. Chapters II-VII are designed primarily for the analyst who is using the ODSAS methodology/information system to aid in solving OPMS problems. Chapter IX, which contains catalogued runstreams, is of interest to both the analyst and the programer. Chapters VIII, X, XI, and XII contain the detailed information necessary to maintain and/or modify the computer programs. This latter information, of primary interest to the programer, is provided in the form of extensive intraprogram commentary in the program listings, file descriptions, and control runstream listings.

d. Two proprietary software packages are utilized in the information system: (1) the UNIVAC Functional Mathematical Programing System (FMPS) and (2) the Marshall Information Retrieval and Display System (MIRADS). Both software packages are documented in separately published manuals. Access by the user to the FMPS programer's reference manual (Reference 1), MIRADS Implementation Manual (Reference 2), and MIRADS User's Manual (Reference 3) is also assumed.

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER II
PREPARATION FOR ODSAS PROCESSING

1. Purpose. - The purpose of this chapter is to provide a description and an identification of the types of input and the user decisions required to use ODSAS.

2. Types of Input

a. There are five general types of input required in the system:

(1) Force requirements.

(2) Management policies on utilization ratios for preferred specialty pairings.

(3) Management policies on tour lengths for preferred specialty pairings.

(4) Population and attrition/promotion rate data applicable to the reference starting population.

(5) System parameters.

Each of these types of input is described in subparagraph c, below.

b. Source of Data Types and Responsibility for Accuracy. - Data to be input to the ODSAS system are collected from numerous Department of Army personnel agencies. The respective agencies are responsible for providing current and accurate data as indicated in Table II-1.

c. Description of Input, by Type

(1) Force requirement data for each type position (e.g., infantry battalion commander) is extracted from the PERSACS tape file. Data elements for each type position include:

(a) Authorized grade

(b) Primary specialty

TABLE II-1, Data Input Responsibilities

Data	Source of data	Responsible organization
Force requirements	PERSACS	ODCSPER (Authorizations Division)
Utilization ratios	MILPERCEN	Officer Personnel Management Directorate, MILPERCEN
Tour lengths	PERSACS	ODCSOPS (Force Accounting Systems Division)
Population and rates	Officer Master File	Population - Officer Personnel Management Directorate, MILPERCEN
	Automatic Interaction Detector-Officers (AID-O) and Central Integrating Model-Officers (CIM-O)	Rates - ODCSPER (Officer Division)
	RCS DCSPER 403	- ODCSPER (Officer Division)
Parameters	MILPERCEN	Officer Personnel Management Directorate, MILPERCEN

(c) Alternate specialty

(d) Effective date (date the position was/will be authorized)

(e) Termination date (date the position will be terminated, if any)

(f) Number of officer spaces authorized

The aggregate requirements by grade, primary specialty, and year are computed in the initialization phase by checking each PERSACS record for grade and primary specialty and then, after determining if the PERSACS termination date is the same as or later than the year of interest (for example, if $T_{\emptyset}$--the starting point for the network--is

input as the end of fiscal year (FY) 1977, and the termination date is 1977 or after), the number of officer spaces authorized is included for the appropriate grade, specialty, and year.*

(2) Data records on management policies are input, on cards, by preferred specialty pairs. For each of the approximately 600 preferred specialty pairs, the utilization ratio of the specialty pair and the tour length of the primary specialty are required for the grades of COL through MAJ (utilization and tour length for CPTs with more than 8 YOS are the same as for MAJs; CPTs with 8 or less YOS and LTs have repetitive assignments in the primary specialty). Each card contains the following information:

(a) Primary specialty number

(b) Preferred alternate specialty number for the primary specialty of (a), above

(c) Utilization ratio for COLs in the primary and preferred alternate specialty

(d) Tour length for COLs in the primary specialty

(e) Utilization ratio for LTCs in the primary and preferred alternate specialty

(f) Tour length for LTCs in the primary specialty

(g) Utilization ratio for MAJs in the primary and preferred alternate specialty

(h) Tour length for MAJs in the primary specialty

A preferred specialty pair is required in each of the field grades. This assures a source of officers with specialties "m" and "n" to meet future requirements in those specialties.

(3) The starting population for all grades, 2LT through COL, must be described by a beginning year of service (BGNYOS)

---

*The ODSAS methodology (linear programing applied to a network flow problem) envisions a network constructed to represent a span of time, in yearly increments. T_0 (time zero) is a base year, T_1 is 1 year later, and so on to T_N--an arbitrary number of years in the future. For a more detailed description see Chapter II of the ODSAS Study Report (reference 4).

indicating length of service of the most junior officer in that grade, and an ending year of service (ENDYOS) indicating the length of service of the most senior officer in that grade. A population, an attrition rate that includes promotion to the next higher grade, and an attrition rate that does not include promotion to the next higher grade are required for each year of service in the interval BGNYOS through ENDYOS.

(4) Input parameters define the number of authorized OPMS specialties and the number of years in the projection period.* The total budget-authorized strength at $T_Ø$ for each officer grade is also required. Additionally, if any or all of the field grades are to be segmented, then the segmenting instructions must be specified along with the additional input associated with segmentation (i.e., designation of primary specialties to be included in the first subsegment, upper bounds for the primary specialties and limits on degree of fill in alternate specialties).

3. <u>User decisions</u>. - There are four types of decisions that the user must make in order to determine appropriate input values for parameters, or evaluate if resulting output from the initialization phase is appropriate.

a. <u>Problem Segmentation</u>. - The first decision concerns parameter values that determine the problem size for unsegmented processing (number of years in the projection periods (NYRS), total number of authorized OPMS specialties (NSPEC), and total number of preferred specialty pairings (NPREF)). An estimate of the LP problem sizes can be determined by procedures described in Appendix E of the ODSAS study report. Given the resultant problem size estimates, the user can then decide whether the segmentation-within-grade option should or must be selected. For example, if the problem size exceeds the computer system capacity, the decision could be to decrease the parameter values and run unsegmented, or to segment

---

*The ODSAS is designed to consider as many as 5Ø OPMS specialties and up to a 9 year projection period. These are design limitations; however, capacity limitations of the present computer hardware and software restrict utilization of the full design capability. The degree of restriction depends upon the number of authorized OPMS specialties and/or the number of preferred specialty pairings and/or the desired length of the projection period.

the processing of the field grade segments (the largest ones).* The sensitivity analysis on the segmentation-within-grade option (see Chapter V of the ODSAS Study Report) should be considered in arriving at a decision on problem segmentation. The user should be cognizant of why and how the solutions can vary between the segmented-within-grade and the unsegmented options.

b. Selection of Specialties in the Subsegments, and Related Optional Limits. - The second decision, contingent upon selecting the segmentation-within-grade option (first decision), consists of selecting the specialties to be allocated to the two subsegments. Furthermore, for those primary specialties in segment 1, an upper bound on the total authorized strength must be established. There is also an optional input for specifying limits on the degree of fill in each alternate specialty in segment 1 and a decision must be made whether to employ this option. As stated in Chapter V, Sensitivity Analysis, of the ODSAS study report, the choices are critical in terms of comparability of answers to the unsegmented processing mode, therefore, this decision and use of the optional limits should be carefully considered.

c. Appropriateness of Rates. - The third decision relates to the appropriateness of the computed weighted average attrition and promotion rates generated from user input in the initialization phase. An underlying assumption used in the rate computations is that rates derived from past experience with promotion and attrition by grade and year(s) of service (YOS), are valid predictors of future promotion and attrition. If the user has some reason to believe that past conditions might not hold in the future, then a decision is required on whether the rates should be changed (and by what amount) or if the solution, with the original rates, should be interpreted by manually applying the revised rates to the solution values for flows in the network.

d. Selection of a Starting Date. - The fourth decision relates to the choice of a date representing $T_{\emptyset}$, the starting point of the network. Essentially, any date can be specified as the starting

---

*In the operational testing of ODSAS in March 1976, as an individual LP problem exceeded 6000 rows, the probability that the UNIVAC hardware and software would not accommodate the problem increased dramatically (primarily because of insufficient core memory), and the decision was made to utilize the segmentation-within-grade option for processing the LTC and MAJ segments.

date, with the decision influenced by the years to be included in the projection period, and the force structure to be specified. Normally the first day of the fiscal year following a desired FY is a logical date for $T_\emptyset$, since the strength of that date reflects the objective end strength of the desired FY (e.g., if the desired $T_\emptyset$ strength should be that for the end of FY 78 (i.e. 78Ø93Ø), then the starting date ($T_\emptyset$) selected for computation of force structure requirements should be 781ØØ1).

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER III
AUTOMATED INFORMATION SYSTEM

1. Purpose. - The purpose of this chapter is to describe the overall design of the automated information system, and how that design was implemented via commercial software (FMPS), software from other government agencies (NASA's MIRADS), and original programs written at CAA.

2. System Design

a. General. - In August 1975, the ODSAS Study Advisory Group approved the methodology described in Chapter II of the ODSAS Study Report and directed that the methodology be incorporated in a computer-based information system consisting of computational and data processing components and associated data elements. The following overall design concepts were used in developing the automated system:

(1) Divide the system functionally such that the capability to specify and solve the linear programing problems be implemented in components separate from supportive activities (e.g., editing of data).

(2) Utilize high-speed computer disc storage devices for input and output of data.

(3) Retain the solutions in a machine-readable form for analysis--with computer printing to be on a selective basis.

Having established the overall design concepts, the functional divisions were identified and the appropriate computer programs to accommodate them were developed. These programs (source listings in Chapter XII of this manual) were combined to become the ODSAS system--a system that the user could control easily.

b. Procedural Functions Included. - To implement the ODSAS methodology on the UNIVAC 11Ø8 computer, applications programs were developed, or incorporated, for the following functions:

(1) Computation of attrition and promotion rates for each grade.

(2) Creation of edited input data files for all system segments.

(3) Generation of linear equations for each segment.

(4) Solution of the linear equations--UNIVAC's Functional Mathematical Programming System (FMPS) level 6.R1B, a standard program product that includes procedures for solving linear programing (LP) problems, was selected for this function.

(5) Specification of FMPS procedures to control the processing while obtaining a solution (e.g., specifying actions to take upon encountering error conditions, or identifying information to be output).

(6) Linkage of one segment or subsegment to another (e.g., updating files to reflect solutions of previous segments).

(7) Interpretation of linear program solutions and production of management reports.

c. System Phasing. - The automated information system developed for ODSAS is comprised of an initialization phase and a processing phase. In the initialization phase, the functions listed in subparagraphs b(1) and (2) above are accomplished. The initialization phase is executed only once. The processing phase accomplishes the functions in subparagraphs b(3) through b(7). The processing phase is repeated for each grade segment or subsegment specified by the user. Segmentation-within-grade, if selected, requires modification to one input file containing user-supplied segmentation instructions. Based upon those segmentation instructions, the appropriate linear equations are generated for the grade segment or subsegment specified.

(1) The Initialization Phase. - At Figure III-1 is a system flow chart of the initialization phase. As shown, there are four user-supplied input data files needed for the ODSAS file creation, data editing, and rates computation procedures. The results of those procedures are output as four computer disc files and two printed reports.

(a) Initialization Phase Input. - The input data comes from three sources. Those sources and format of all the data contained in the input files are described in Chapter IV of this documentation.

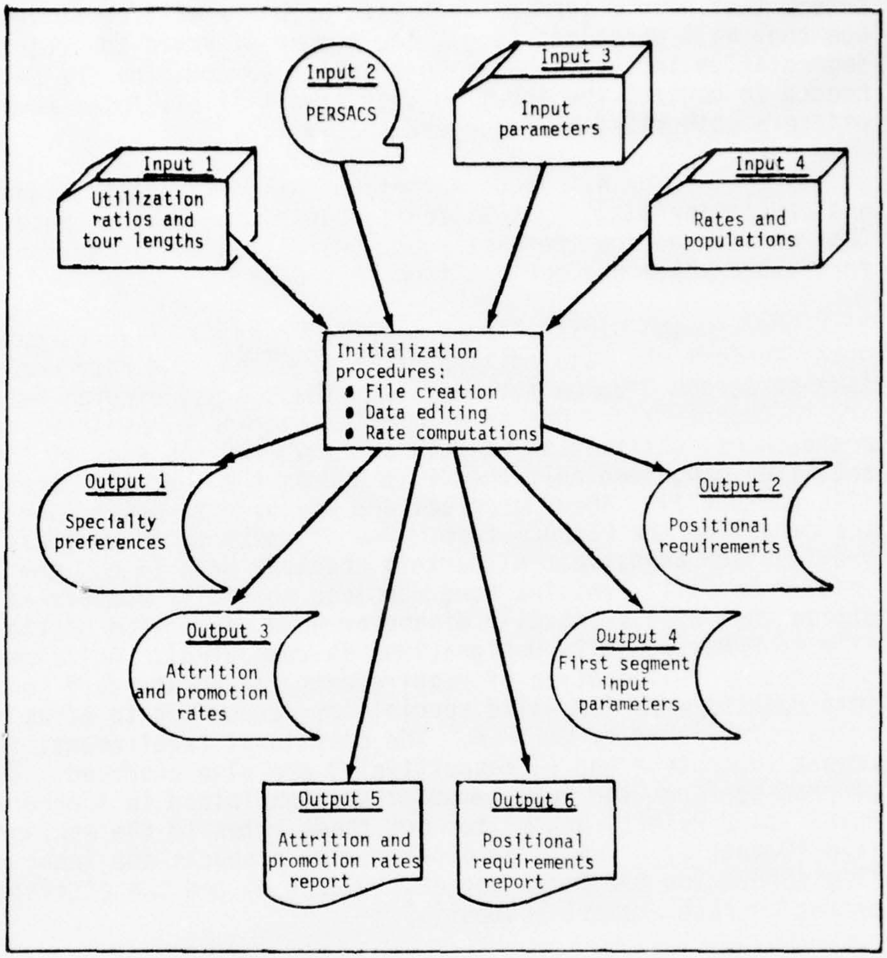


FIGURE III-1, ODSAS Automated Information System, Initialization Phase

1. The first input file (labeled Input 1 on Figure III-1) contains the policy (official or test) on the utilization ratios and tour lengths (in all grades) for preferred specialty pairs.

2. Input 2 is the Personnel Structure and Composition System (PERSACS) data file of present and future requirements by grade and specialty.

3. Input 3 is the specification of system parameters that help determine what size problems will be solved and how they will be solved (e.g., the number of years to project and segmentation instructions). This file also contains the data values needed to control the input to the network (i.e., total number of officers authorized by grade and specialty).

4. Input 4 contains historical attrition rates and population data, by YOS, applicable to the officer population that existed during the past year. Attrition and promotion rates for future years are derived from this data.

(b) Initialization Procedures. - Five computer programs perform the data editing, file creation, and rate computations (see paragraph 3 below for detailed flow charts; program narratives are in Chapter XI). One program edits user-supplied input data on preferences, utilization ratios, and tour lengths and, if the data satisfies programed edit checks, produces the specialty preferences file (Output 1). Three programs are employed to select and edit the data from the PERSACS tape. One of these three programs also provides for adjustment of certain obsolete data (i.e., the PERSACS input file still contains some obsolete specialty numbers--a condition that will eventually disappear when conversion of the PERSACS file to OPMS specialty designations is completed). Adjustment is effected by reallocation of requirements for nonstandard specialties into requirements for valid specialties, according to predefined rules established by ODCSPER. The positional requirements file and report (Outputs 2 and 6, respectively) are also produced. The fifth program performs the rate computations (explained in Appendix D of ODSAS Study Report) and writes out these rates to the appropriate file (Output 3). The third program also produces the input parameter file needed for the first segment (Output 4) and the attrition and promotion rates report (Output 5).

(c) Initialization Phase Outputs. - As a result of the initialization procedures, output disc files (numbered 1 through 4 in Figure III-1) are produced. The files are described in Chapter VIII, Computer Disc and Tape File Descriptions, and the reports in Chapter V, ODSAS Printed Output.

1. Output 1, the specialty preference file, contains the utilization ratios and tour lengths of all preferred specialty pairings, for all grades, arranged within grade and specialty.

2. Output 2 contains the positional requirements (i.e., requirements derived from the PERSACS input for all grades and specialties in the years of the projection period).

3. Output 3 contains the computed attrition and promotion rates for each grade, per year of the projection period.

4. Output 4 contains all the parameters and rates needed as input for the processing of the first segment. The input files for the subsequent segments are produced in the processing phase as they are needed.

(d) Reports. - The two printed reports (Outputs 5 and 6 of Figure III-1) are for verification and retention by the user. The reports display the results of computation of the attrition and promotion rates and the requirements by grade, specialty, and year. Examples of the reports are shown in Chapter V.

(2) The Processing Phase

(a) General. - Figure III-2 is a system flow chart of the processing phase. The processing phase is comprised of five major activities, the five blocks indicated by the dashed lines in Figure III-2.

1. Major activity 1, the matrix generator, produces the LP equations in FMPS format.

2. Major activity 2, FMPS solution, solves the equations and provides selected solution data for subsequent use.

3. Major activity 3, data base creation, creates the input files and loads them on to the data base.

4. Major activity 4, an on-line inquiry system, permits the user to evaluate system output during processing.

5. Major activity 5, linkage, connects one segment or subsegment to the next, to provide continuity of processing. The processing phase is done at least five times (once for each grade--COL through LT). If the segmentation-within-grade option is selected for any of the field grades, up to three additional iterations of the processing phase would be required (one for each grade segment).

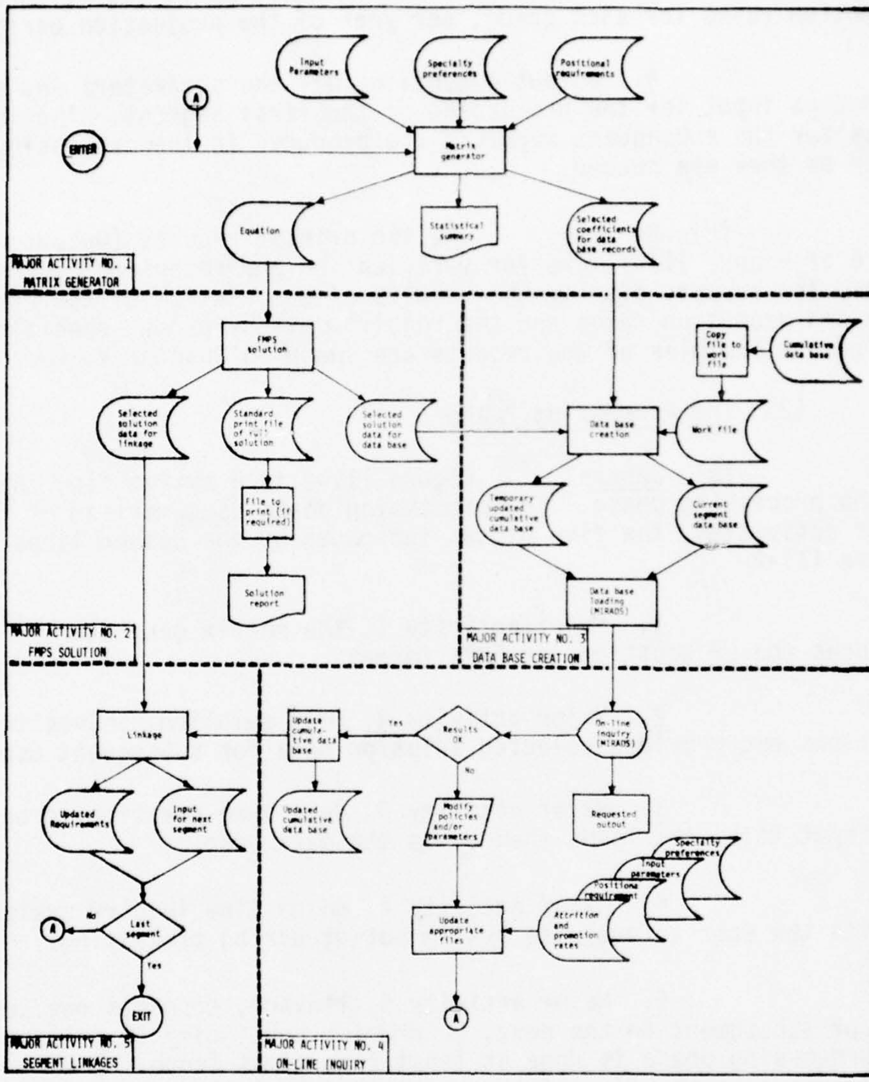


FIGURE III-2, ODSAS Automated Information System, Processing Phase

(b) Description of Major Activities

1. Matrix Generator. - The matrix generator, major activity 1, is depicted at Figure III-3. Accessing data on files created in the initialization phase, the matrix generator

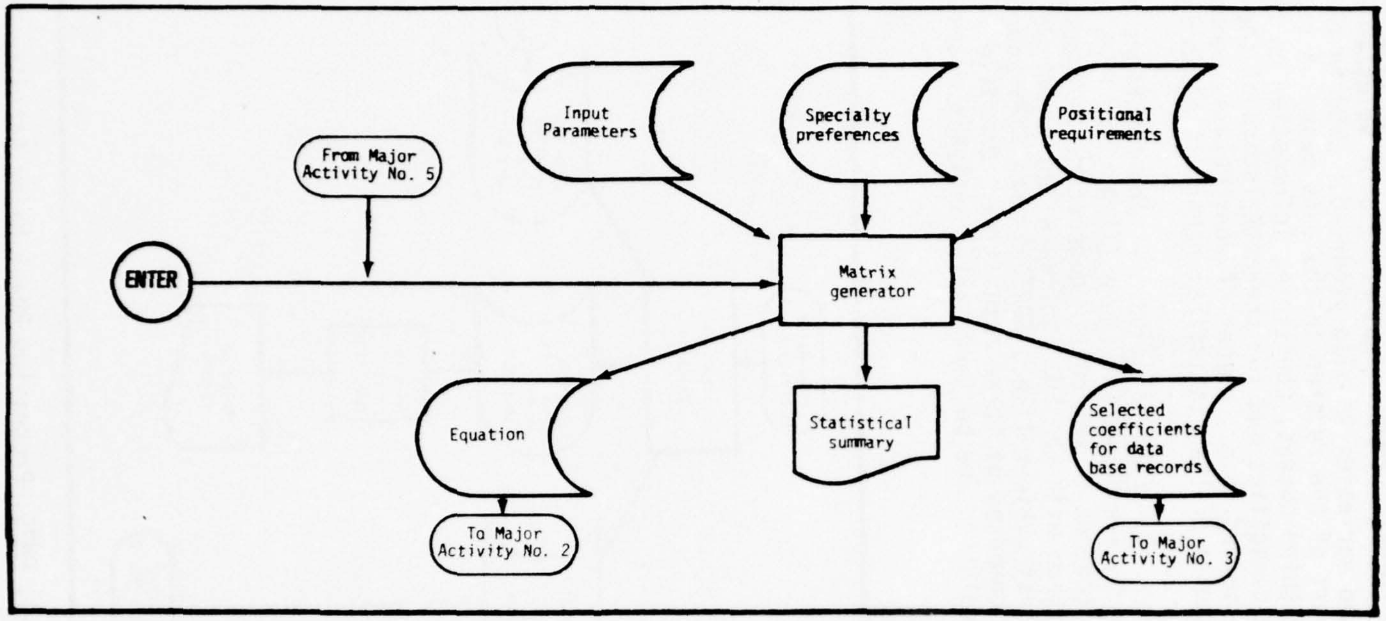


FIGURE III-3, Matrix Generator, Processing Phase Major Activity Number I

programs produce an equation file organized according to the standard format of UNIVAC's FMPS. Another file of selected data on the constraints and variables is also produced. Data in the latter file will become part of the records in the data base. A statistical report is the third output, containing information on the network structure and capacities and the characteristics of the linear program to be solved (e.g., number of constraints, number of variables). An example of the statistical report is shown in Chapter V.

2. FMPS Solution. - The functions of FMPS, major activity 2, are shown in Figure III-4. The FMPS accepts the equation file (output of the matrix generator) as input, and solves the linear program with the FMPS software and a user-defined set of implementing instructions (i.e., FORTRAN-like FMPS source statements). The output is composed of three data files. One file is the standard FMPS printed solution and postoptimality analysis output that can

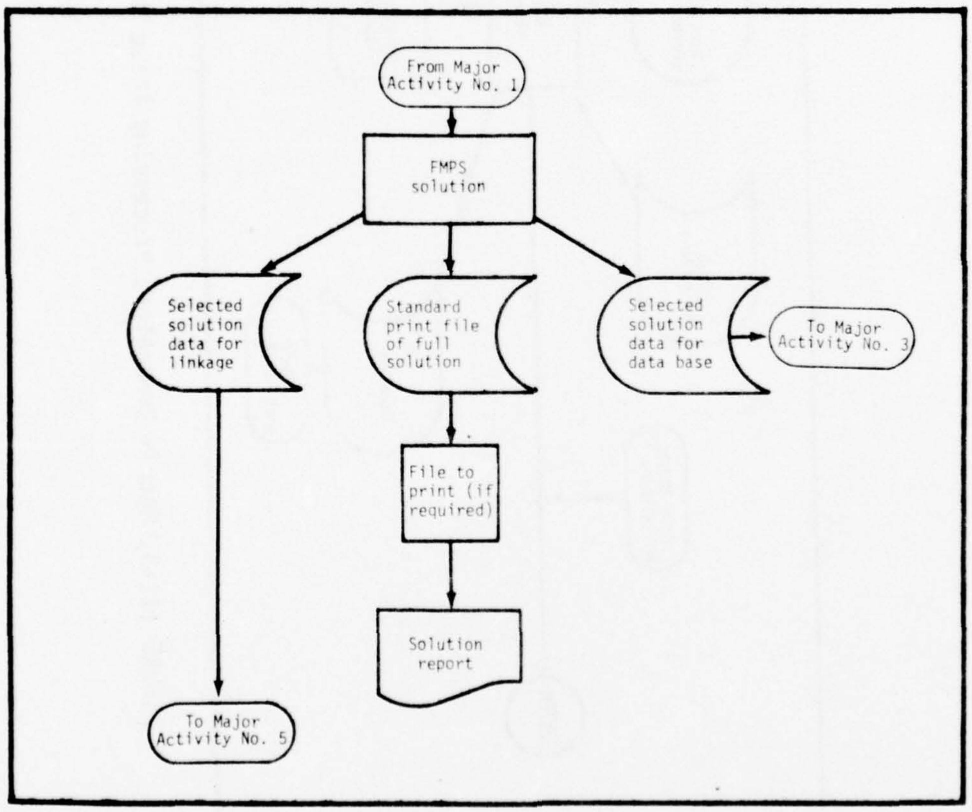


FIGURE III-4, FMPS, Processing Phase Major Activity Number 2

be printed, selectively, on a high-speed printer or analyzed with a text editor via a computer terminal. The other two files contain selected data items on the constraints and variables in the LP problem; one file supplies data to the data base and the other file passes information on filled officer requirements to the linkage activity, so that the requirements in the next grade segment initially reflect only unfilled requirements.

3. Data Base Creation. - The third major activity of the processing phase (Figure III-5) involves accessing information from two of the files produced in the first and second major activities, along with a file of the cumulative results of any previous system segments. The cumulative data base file is first copied to a work-file for two reasons:

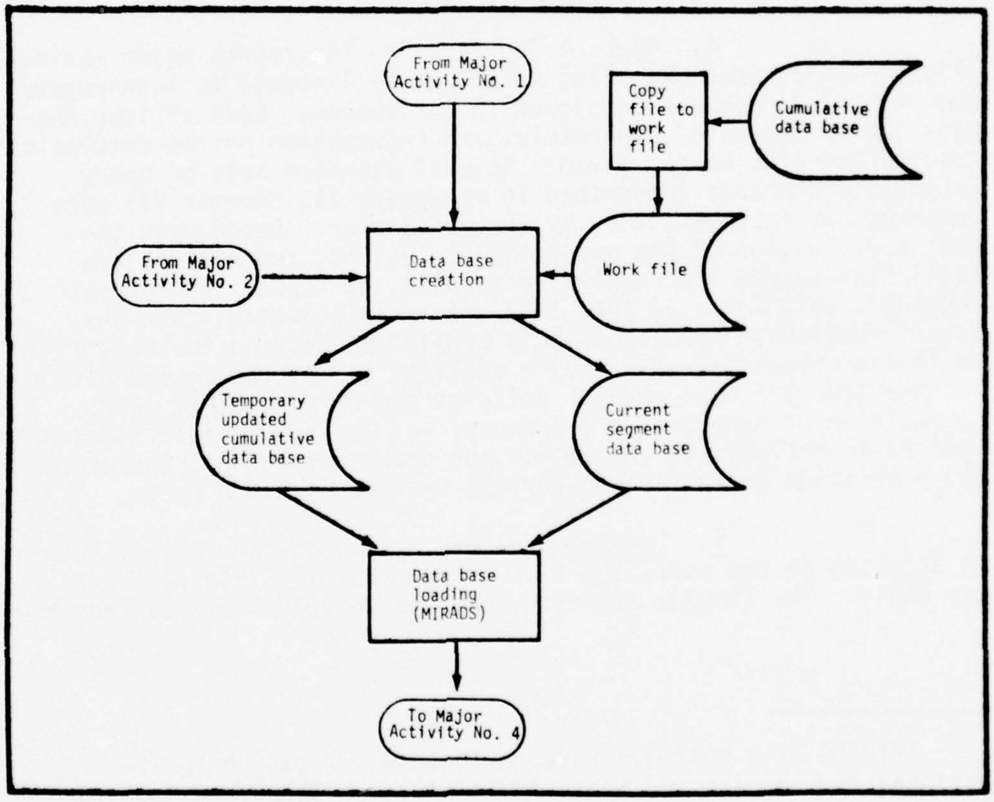


FIGURE III-5, Data Base Creation, Processing Phase Major Activity Number 3

a. If the segment results are unacceptable as determined by the user during major activity 4, then the actual cumulative file up to, but not including, the current segment is not updated, and will be available when the current segment is processed again.

b. The cumulative results, to include the current segment, can be evaluated on the work-file without inhibiting further processing of the system.

The two files from the first and second major activities are combined to produce a data base of information on the current segment. A temporary file (a copy of the cumulative results) is also updated to produce a cumulative data base that includes the current segment results. The Marshall Space Flight Center Information Retrieval and Display System (MIRADS) (references 2 and 3) is used to load the data base and prepare the information for the on-line inquiry conducted in the fourth major activity.*

4. On-Line Inquiry. - The fourth major activity (Figure III-6) involves using MIRADS query language to interrogate the several data bases developed in the system. Each officer segment can be evaluated separately, and information on the cumulative results can also be retrieved. Several standard sets of query language statements (described in paragraph 11, Chapter VI) were prepared for implementation by the ODSAS user. Based upon the user's evaluation of the cumulative solutions, two options are available--accept the cumulative results, or reject the current segment's solution. If the first option is selected, then the actual cumulative results file is updated by copying the work file to it and proceeding on to major activity 5. If the second option is selected, the user changes policies and/or parameters (e.g., composition or number of preferences) and the appropriate files are updated to reflect the change via the update procedure, whereupon the processing phase for the current segment is begun again.

5. Segment Linkages. - Once the current segment is accepted by the user, the fifth major activity (Figure III-7) can begin. The linkage activity uses the solution results from the

---

*MIRADS is a software package that was developed for the National Aeronautics and Space Administration (NASA) by Computer Sciences Corporation for use on UNIVAC 1108 computers and was furnished free of charge to CAA and MILPERCEN.

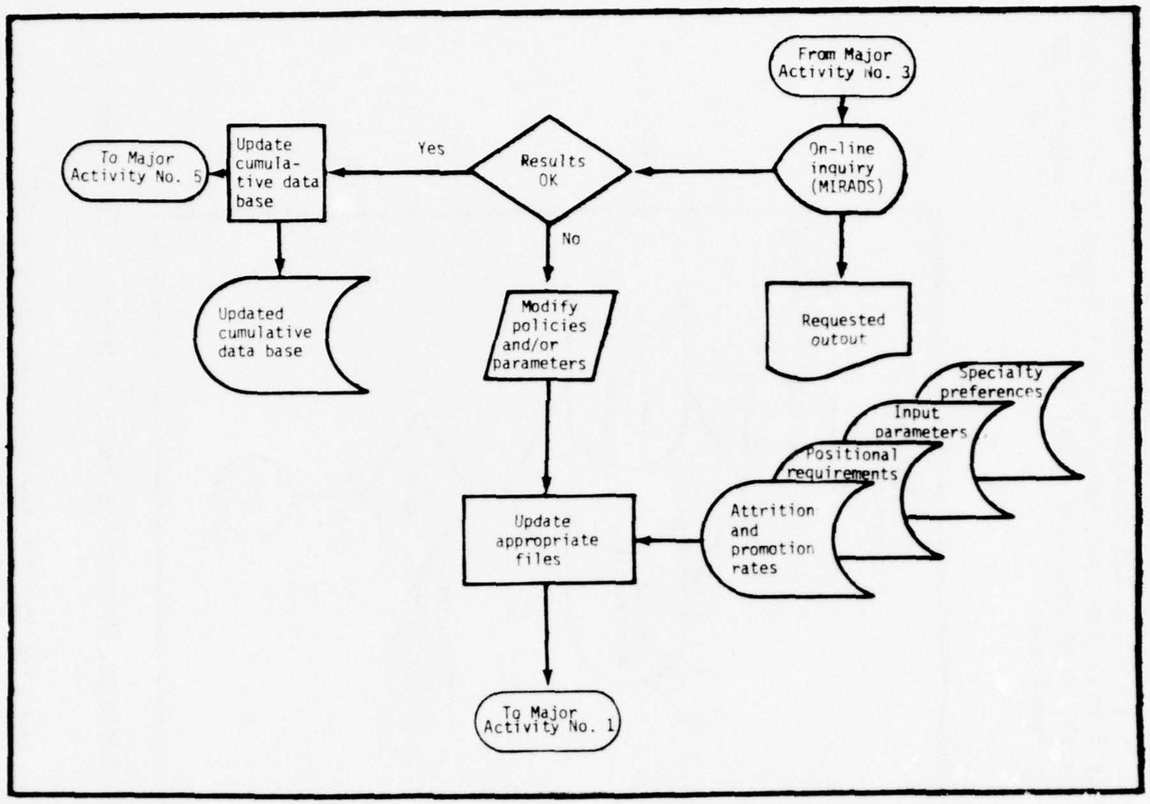


FIGURE III-6, On-Line Inquiry, Processing Phase Major Activity Number 4

current segment and computes how many requirements remain to be filled by subsequent segments. A new input file is created for the next segment in sequence and the requirements file is updated to reflect the unfilled requirements through the current segment. The next step is to begin the processing phase again by performing major activity 1 with the next segment, or subsegment, in sequence. The processing phase is repeated until the final (LT) segment is satisfactorily completed.

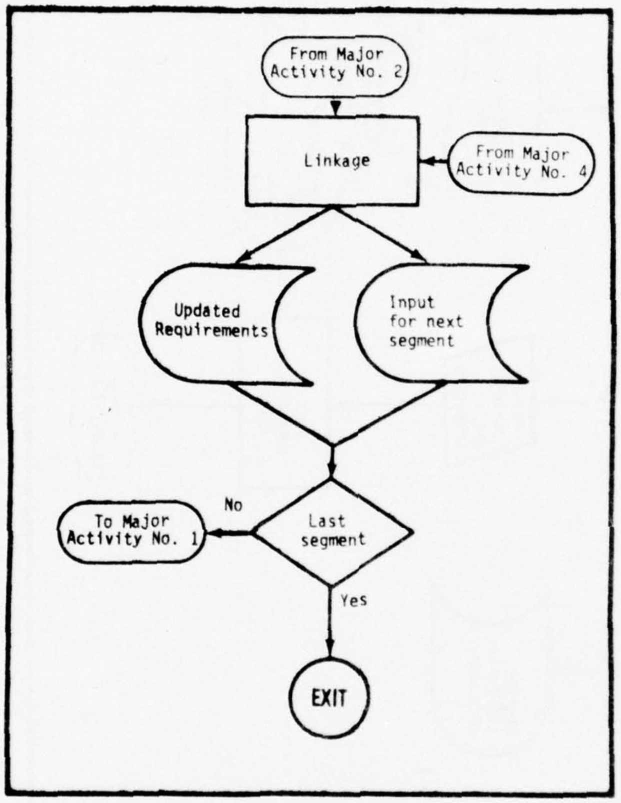


FIGURE III-7, Linkage, Processing Phase Major Activity Number 5

3. Discussion and Flow Charts for Initialization Phase. - The initialization phase is depicted in Figure III-1 as a single procedure block (i.e., Initialization procedures). However, the several functions indicated within that block (i.e., File creation, Data editing, and Rate computations) are actually accomplished by five CAA written programs (SACSEXTRACT, SACSPREPRO, SACSCREATE,

TOURATIOS, and INITIAL) that are structured into three catalogued runstreams (PFCAA.SACS, PFCAA.TOUR, and PFCAA.INITIAL) and one runstream in card form.

a. The latter runstream (illustrated at the top of the flowchart in Figure III-8) is used on a computer system in MILPERCEN

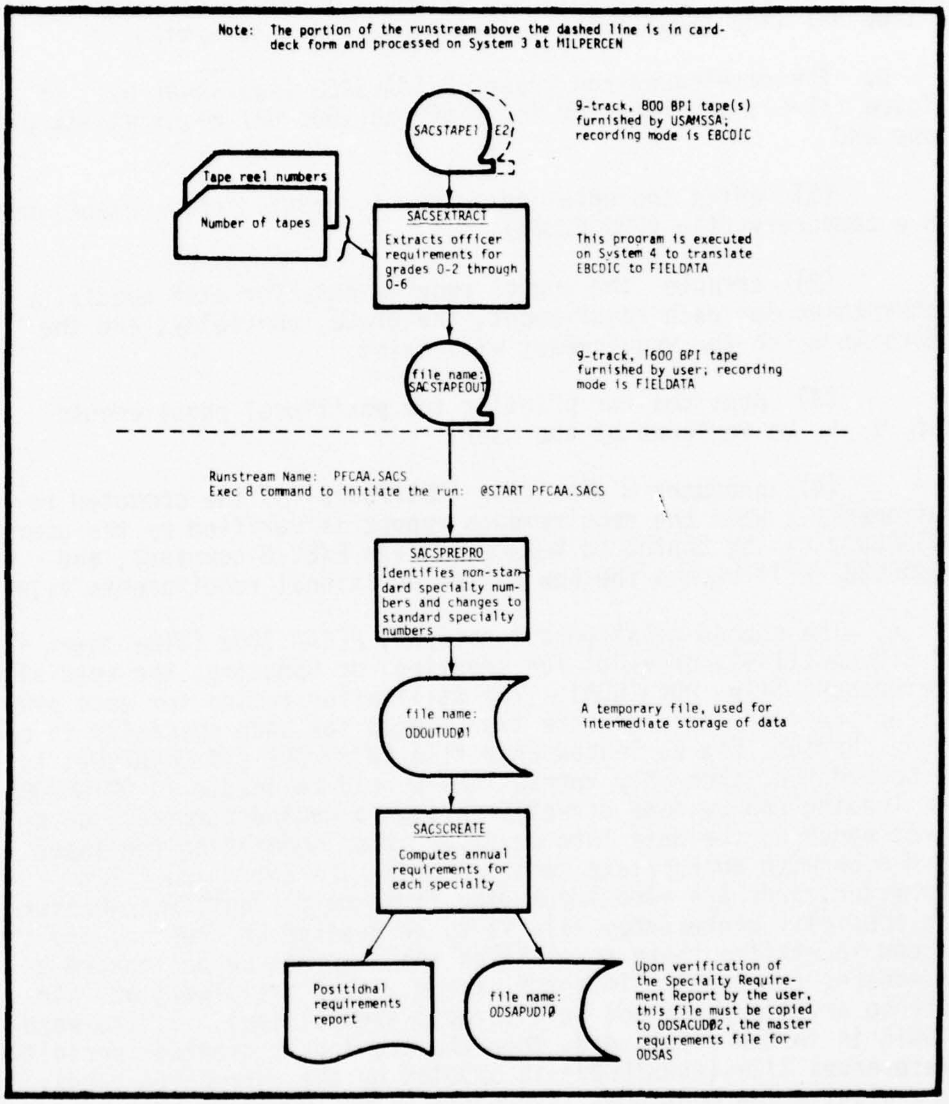


FIGURE III-8, Structure of Runstream PFCAA.SACS

with a hardware tape translation capability (translates 800 BPI EBCDIC tapes to 1600 BPI FIELDATA for use on System 2). The input tapes (SACSTAPE1 and, if required, SACSTAPE2) are furnished by USAMSSA and contain requirements data for all grades. SACSTAPE2 is a continuation of the data on SACSTAPE1 and is required only when there is too much data to be contained in a single tape. The parameter cards input to the SACSEXTRACT program specify the number of tapes to be input and the identifying tape reel number(s). The SACSEXTRACT program extracts COL through LT data and writes it to a 1600 BPI tape (SACSTAPEOUT) that can be read on System 2.

b. The catalogued runstream, PFCAA.SACS (see lower half of Figure III-8), provides for input of the 1600 BPI requirements data tape and

(1) edits the data and writes the edited requirements data to a temporary file (ODOUTUD01).

(2) computes the annual requirements for each specialty by determining for each requirement, the grade, specialty, and the years in which the requirement will exist.

(3) provides for printing the positional requirements report, to be reviewed by the user.

(4) produces a disc file (ODSAPUD10) of the computed requirements. When the requirements report is verified by the user, ODSAPUD10 can be copied to ODSACUD02 (via EXEC 8 command), and ODSACUD02 will become the new master positional requirements file.

c. The second catalogued runstream, PFCAA.TOUR (flow chart is at Figure III-9), provides for creating, or updating, the specialty preferences file, ODRATUD01. The utilization ratios for each preferred specialty pair and the tour length for each specialty in each grade should first be loaded onto file ODTURUD01 (if ODRATUD01 is to be updated, then only corrections should be loaded to ODTURUD01). The loading can be done directly through a demand terminal, or by first punching the data into cards and then submitting the input card deck with appropriate control cards as a batch job. Two parameter cards are also input--the first card identifies whether the specialty preferences file is to be updated or created, and the second identifies those specialties that may not be designated as alternates (currently, infantry, armor, field artillery, and air defense artillery can not be alternate specialties). If the word UPDATE is in the first card, then the previously created specialty preferences file (ODRATUD01) is updated by the correction cards; otherwise, a new ODRATUD01 is created. The TOURATIOS program converts the utilization ratios expressed as decimals to ratios

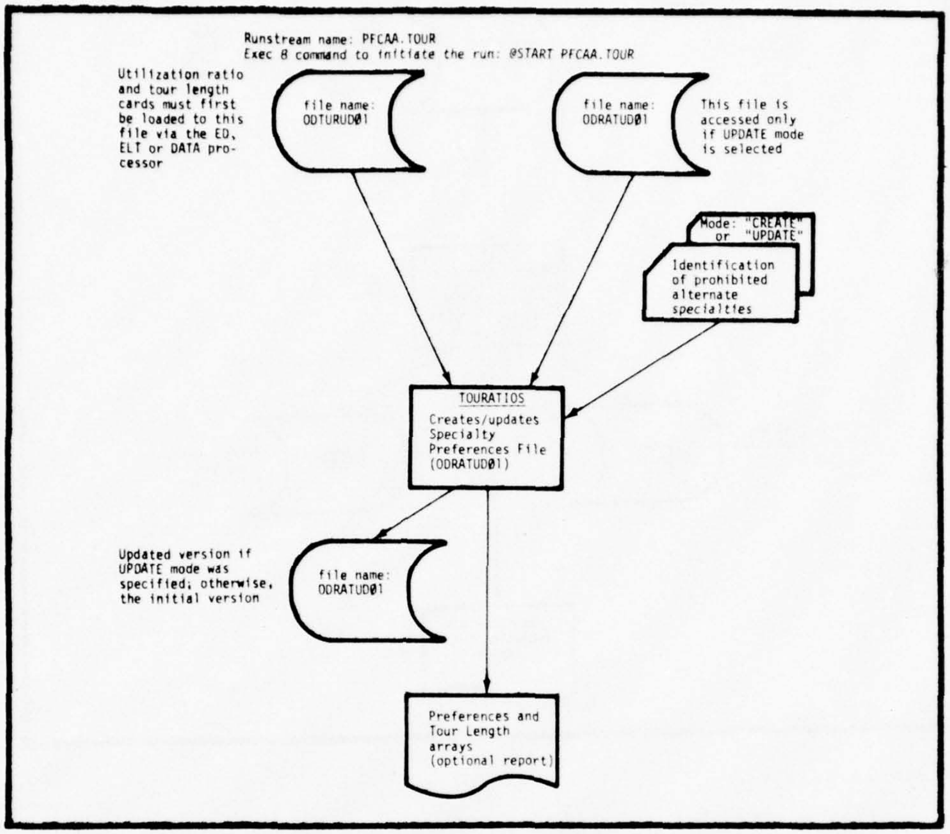


FIGURE III-9, Structure of Runstream PFCAA.TOUR

between two integers (Ø.75 converts to 3:1), and places the ratios (3:1 stored as 31) in an internal array (a 5Ø x 5Ø matrix) that relates primary and alternate specialty pairs. A prohibited alternate specialty is coded uniquely in the internal arrays (utilization ratio of 88), and all other possible specialties (those not specifically mentioned) are coded as 99. Specialties coded 88 and those coded 99 are treated as invalid specialty pairs.

d. The third catalogued runstream, PFCAA.INITIAL (illustrated in the chart in Figure III-10), provides for creating the files of input parameters and rates used in all the system segments. The input parameters for all segments, and attrition rates and population data for a reference population are input via the ODR8SUDØ1 file. The weighted average attrition, promotion, and retention rates (for CPTS with 8 YOS) are computed for a 9-year projection

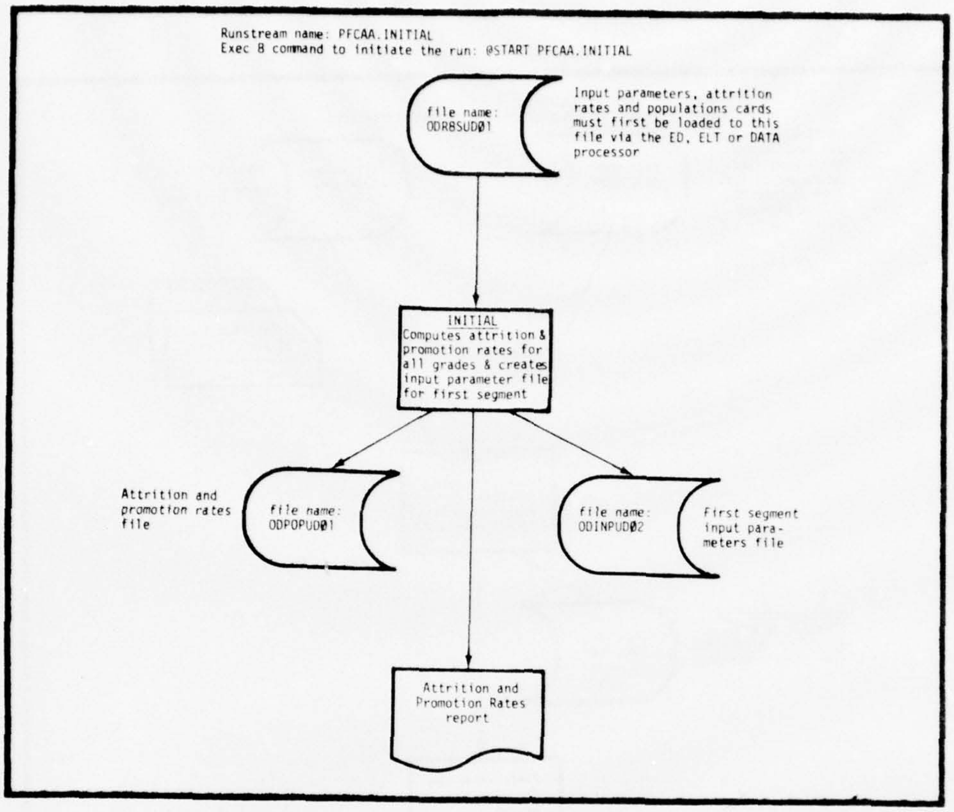


FIGURE III-10, Structure of Runstream PFCAA.INITIAL

period. The weighted average rates are stored by grade and projection year on file ODPOPUDØ1. The calculations for each rate are displayed on the Attrition and Promotion Rates report. The input parameters are assembled in the order in which they will be needed to run each ODSAS segment, and are placed on file ODINPUDØ2. The attrition and promotion rates for grade 6 are also placed in ODINPUDØ2, such that it contains all the data needed to start ODSAS processing. At the beginning of ODSAS processing, ODINPUDØ2 is copied to ODINPUDØ1, and thereafter the latter file is updated at the completion of each segment. Thus, ODINPUDØ2 is the master file that can be used to restart ODSAS from the beginning, without repeating all the initialization procedures.

4. Discussion and Flow Charts for Matrix Generator Activity. - At the system flow chart level (Figures III-2 and III-3 above), the matrix generator is represented by one procedure block; however,

there are actually one main program (an executive controlling program) and 32 subroutines or functions.

a. A system flow chart of the matrix generator activity, which includes the actual data file names (as described in Chapter VIII) used in the EXEC 8 runstream, is shown in Figure III-11. For the input files, the file name ODINPUDØ1 corresponds to the input parameters file; ODRATUDØ1 corresponds to the specialty preferences file; and ODSACUDØ1 corresponds to the positional requirements file,

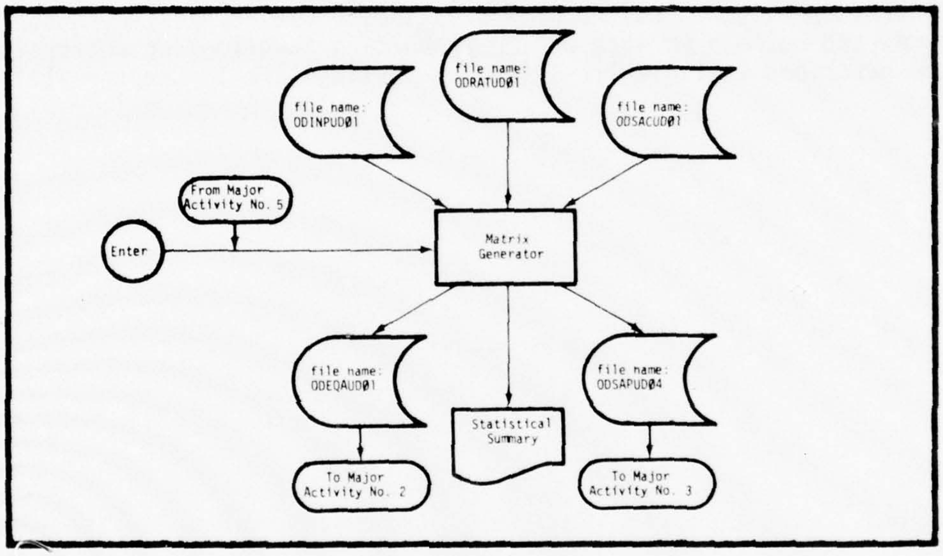


FIGURE III-11, Matrix Generator Flow Chart with Internal File Names Specified

as shown in Figure III-3. For the output files, the file name ODEQAUDØ1 corresponds to the equation file, and ODSAPUDØ4 corresponds to the file of selected coefficients for the data base records, also shown in Figure III-3.

b. The 32 subroutines/functions are organized along lines relating to the input format required by FMPS and the peculiarities of the methodology developed for each grade segment. Thus, there

are four different combinations of subroutines to produce the necessary input file for the system segments for COL, LTC, MAJ, CPT, and LT. The flow charts remain unchanged even when the segmentation-within-grade option is selected for any of the field grade segments; the internal logic in the subroutines and supporting functions cause the appropriate input needed for subsegment 1 or 2 to be generated. The identification of the subroutines, and the calling sequence within each grade segment, for generating the linear equations for the FMPS programs are shown in Figures III-12 through III-15. At the completion of the processing in each major subroutine, control returns to the MAIN program, where the instructions for the next subroutine to be called are located. The numbers at the lower left of the procedure blocks in the flow charts indicate references to minor supporting functions or subroutines. The identifying number, name, and purpose of each of the supporting functions or subroutines are described in Table III-1 (source listings are in Chapter XII).

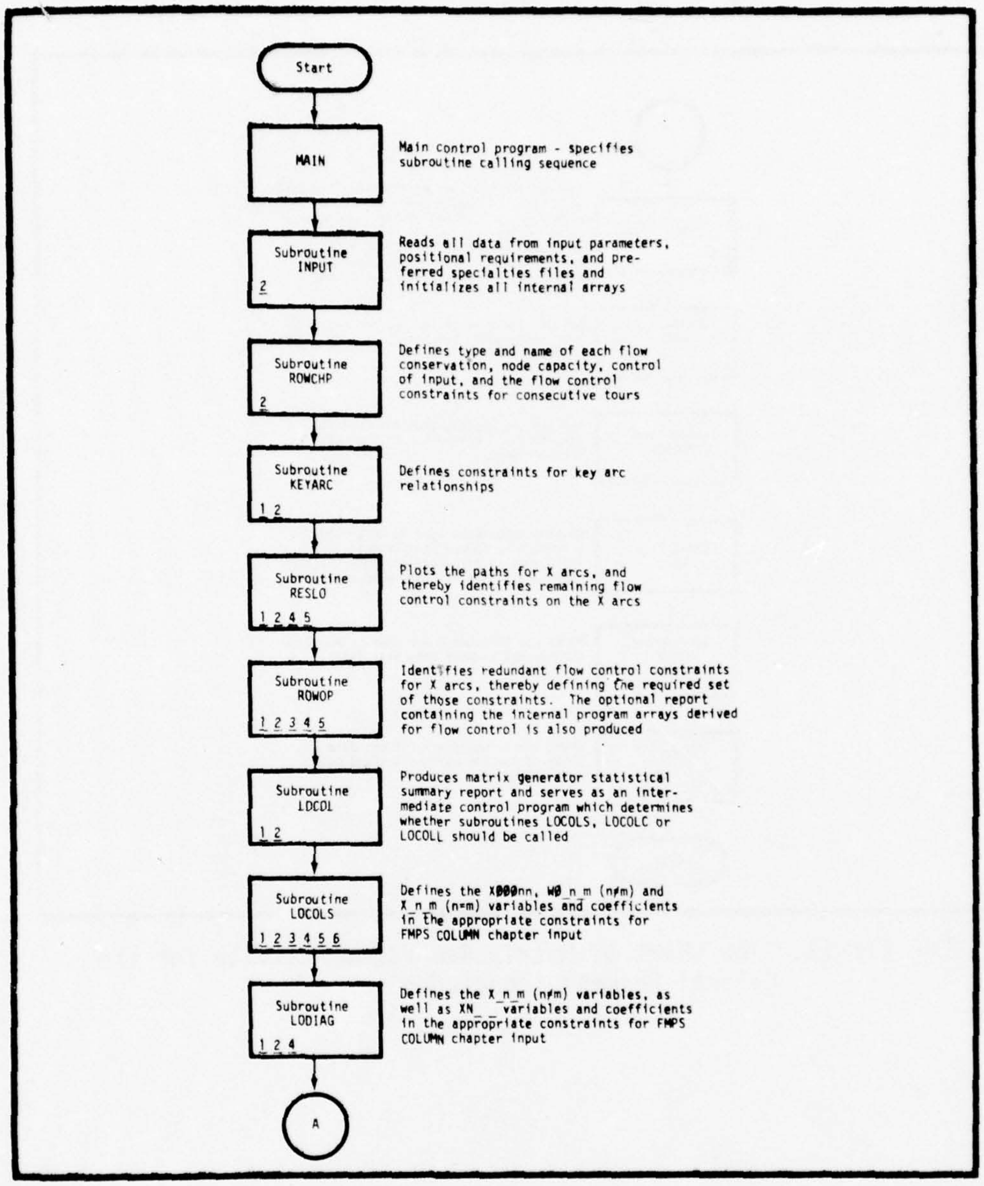


FIGURE III-12, Flow Chart of Matrix Generator Activity for the Colonel Segment (continued on next page)

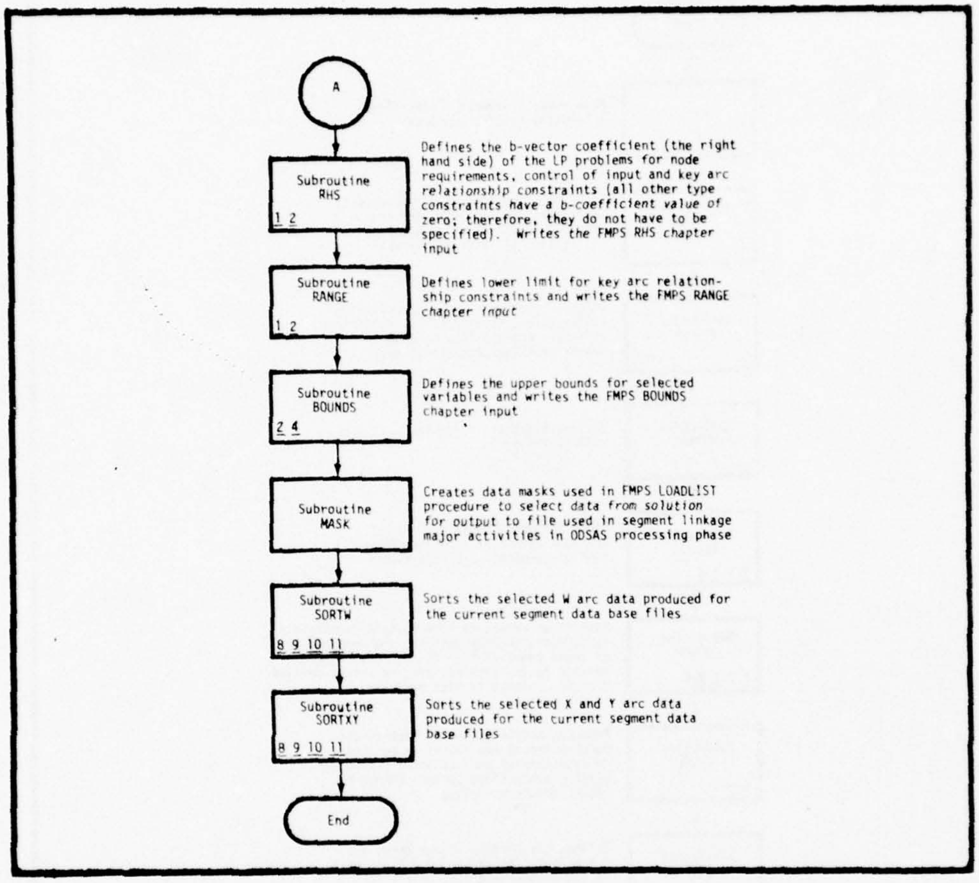


FIGURE III-12, Flow Chart of Matrix Generator Activity for the Colonel Segment (concluded)

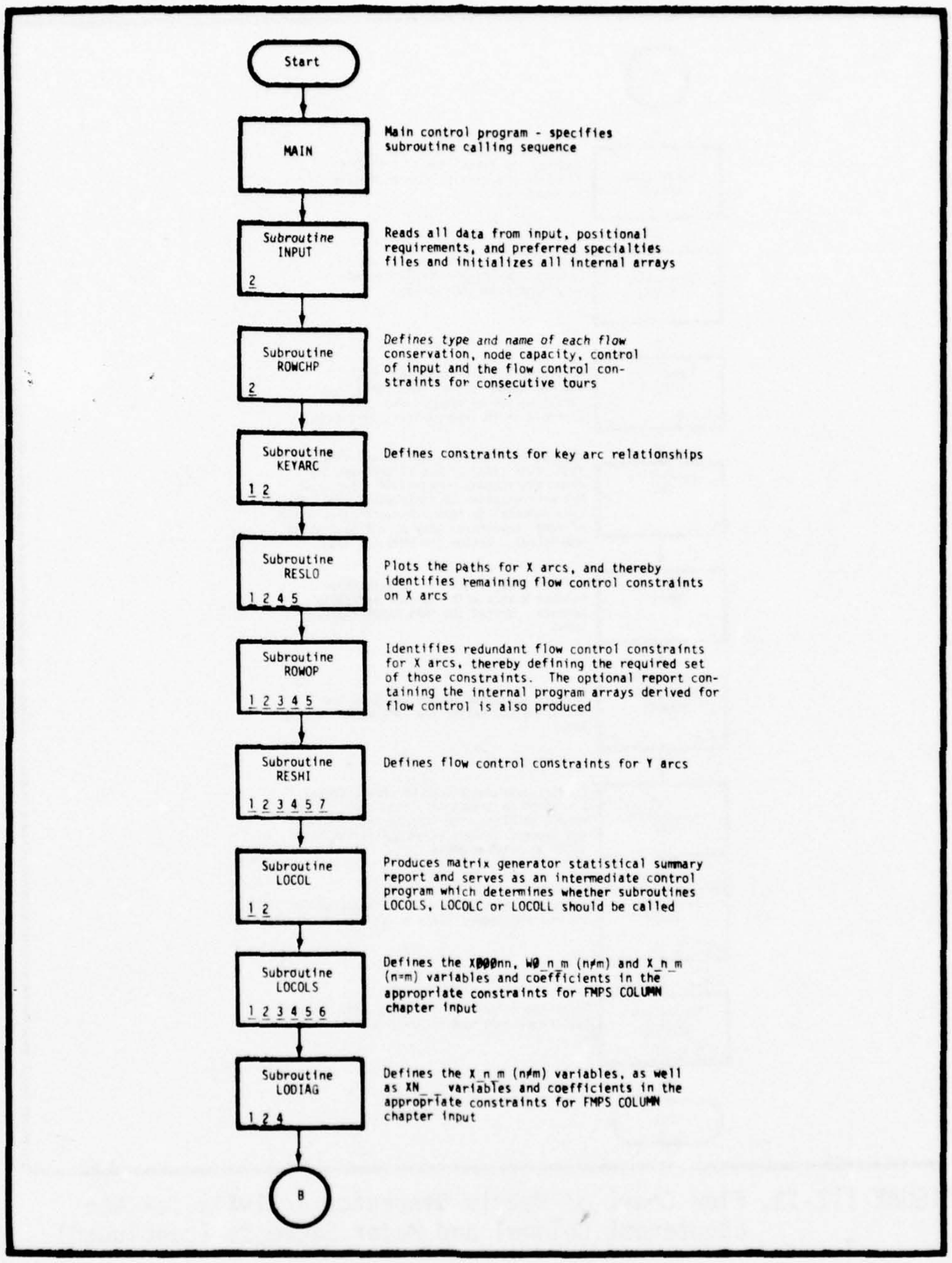


FIGURE III-13, Flow Chart of Matrix Generator Activity for the Lieutenant Colonel and Major Segments (continued on next page)

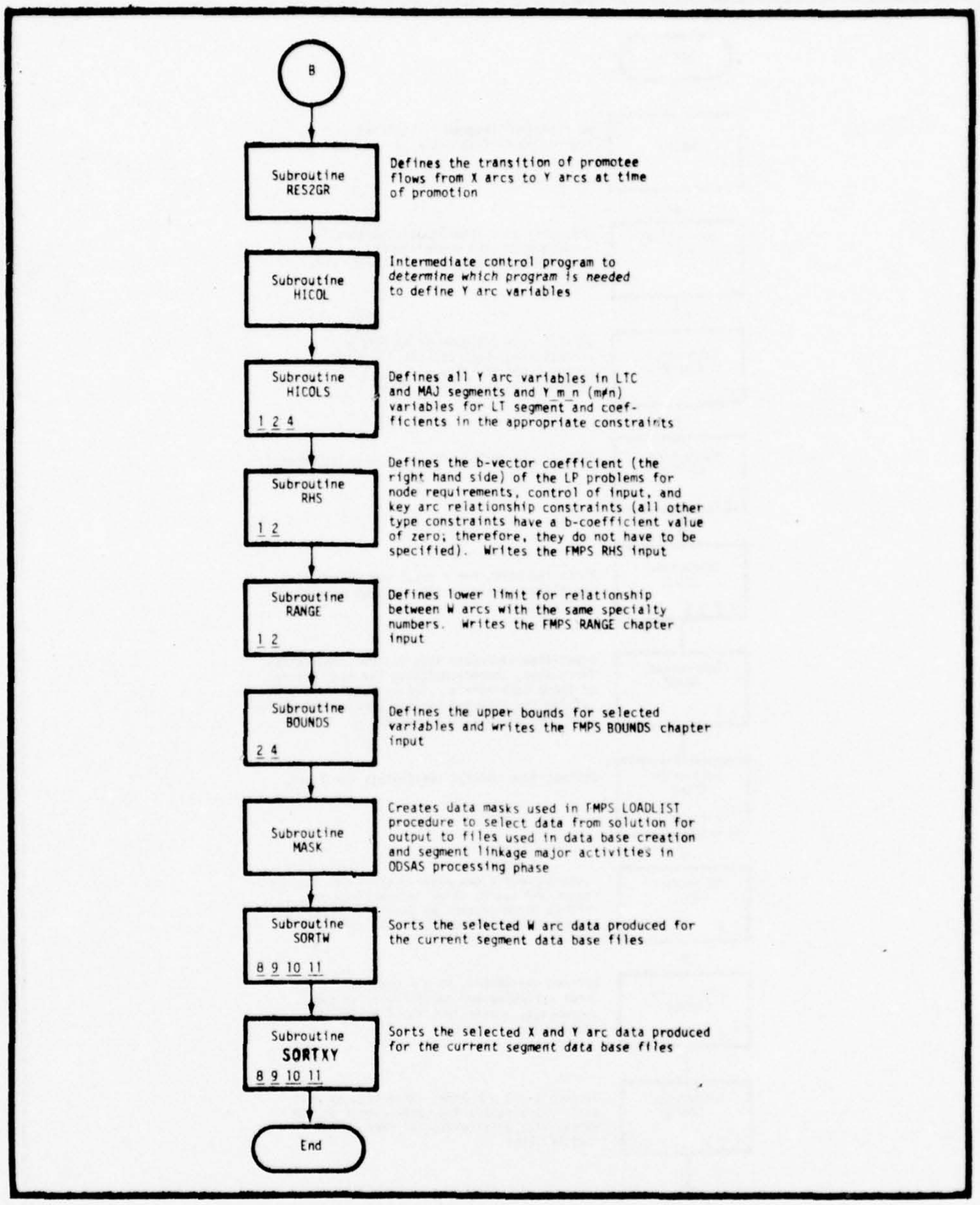


FIGURE III-13, Flow Chart of Matrix Generator Activity for the Lieutenant Colonel and Major Segments (concluded)

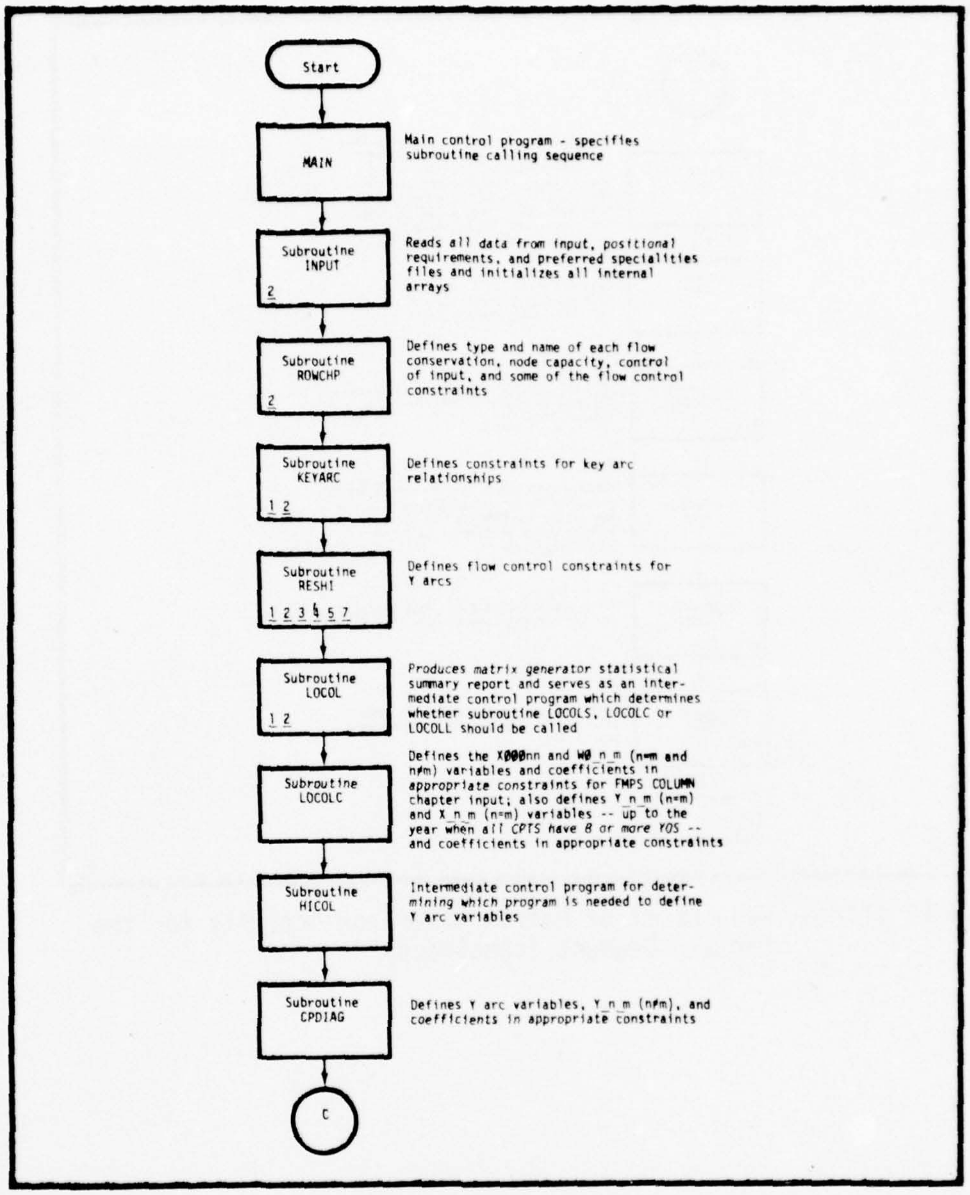


FIGURE III-14, Flow Chart of Matrix Generator Activity for the Captain Segment (continued on next page)

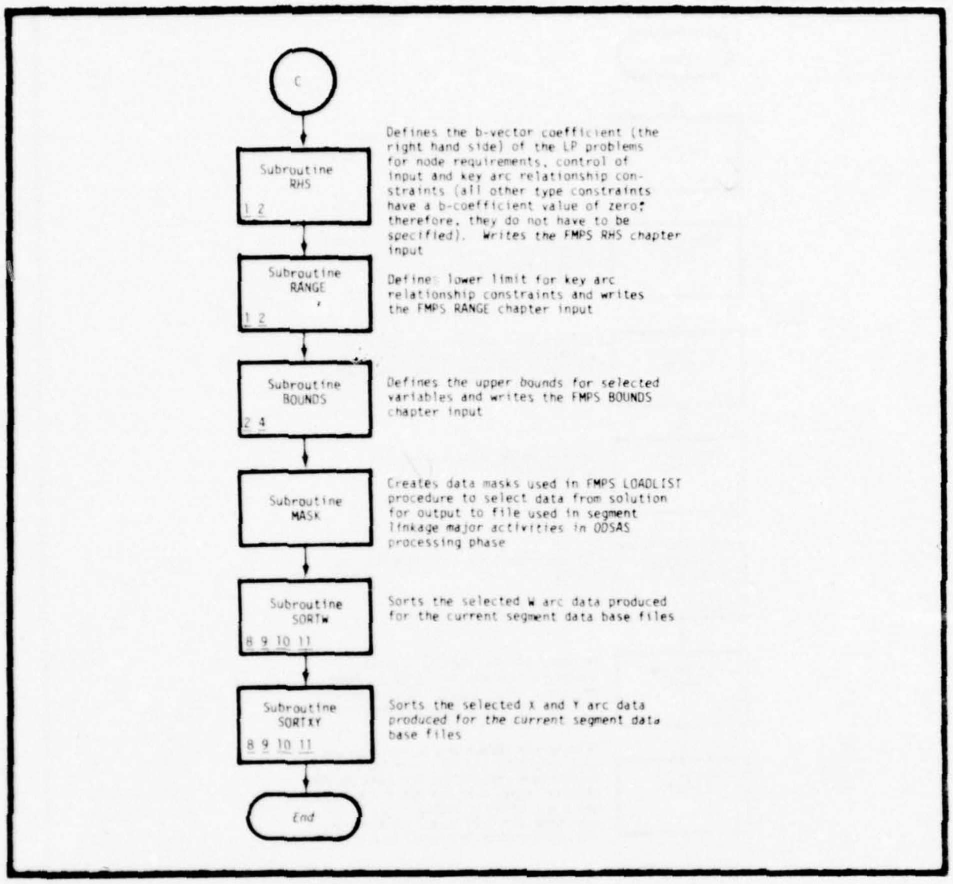


FIGURE III-14, Flow Chart of Matrix Generator Activity for the Captain Segment (concluded)

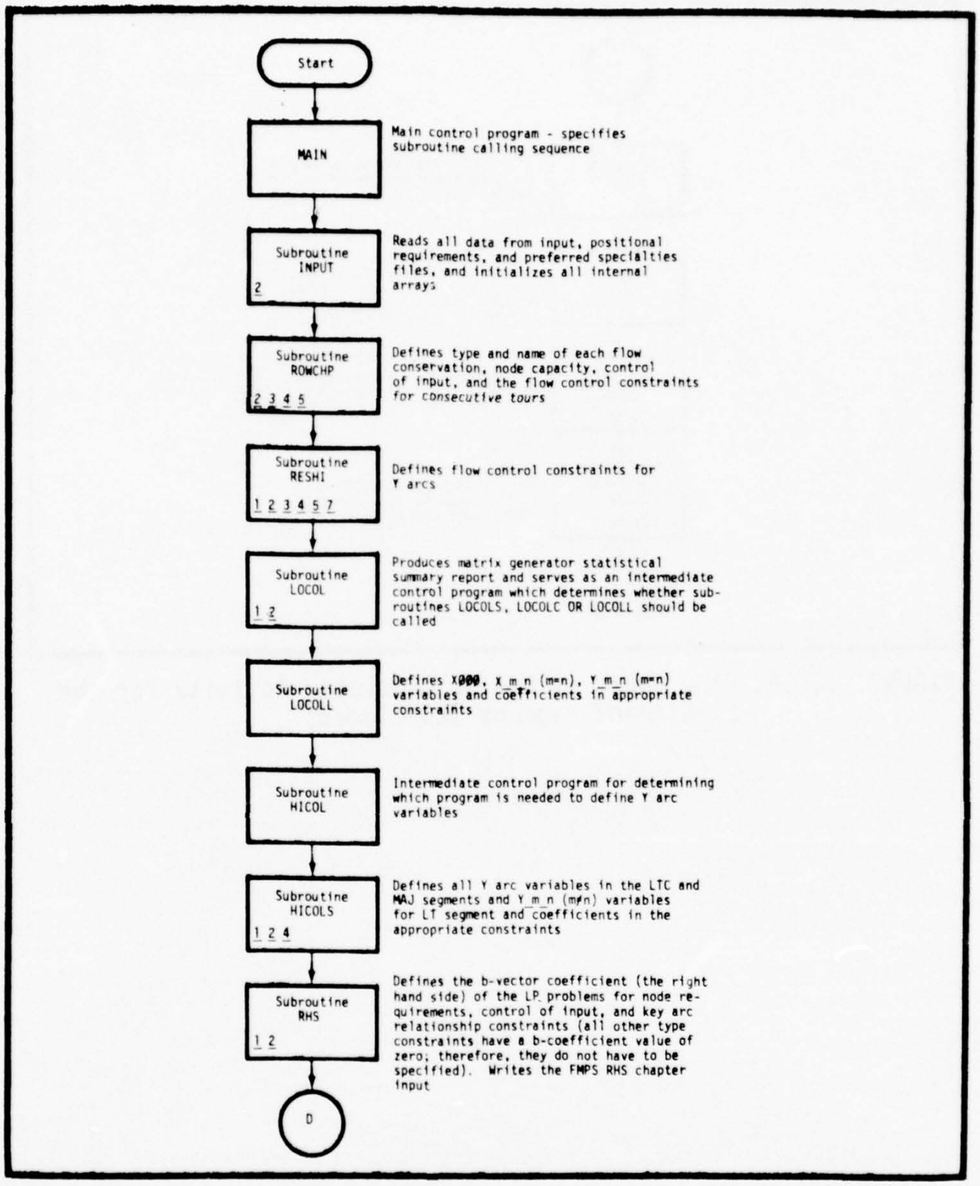


FIGURE III-15, Flow Chart of Matrix Generator Activity for the Lieutenant Segment (continued on next page)

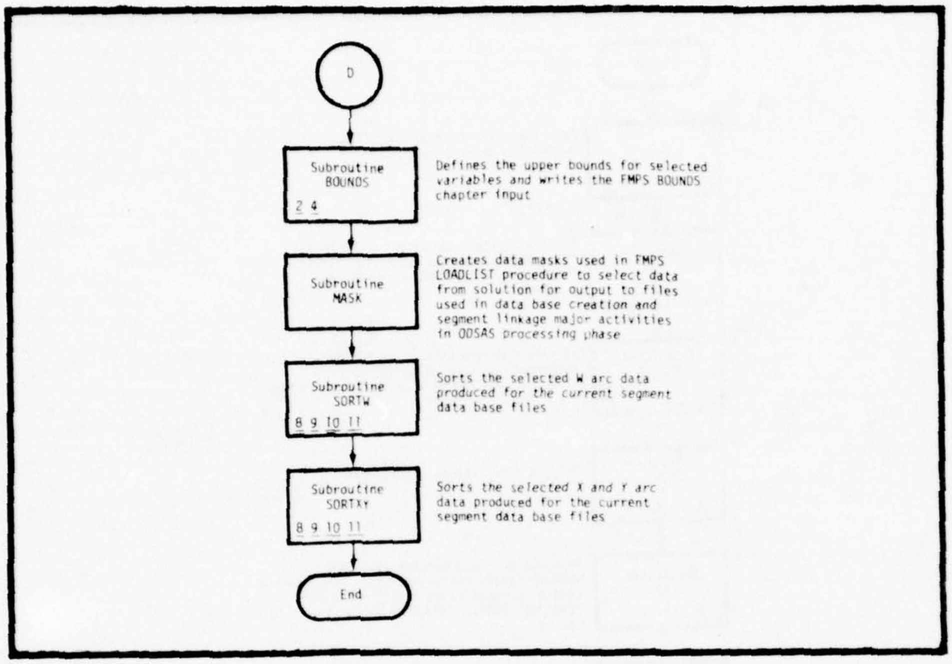


FIGURE III-15, Flow Chart of Matrix Generator Activity for the Lieutenant Segment (concluded)

TABLE III-1, Identification of Minor Supporting Functions or Subroutines Used in Matrix Generator Activity Flow Charts (continued on next page)

Identifying number	Function/ subroutine name	Description
1	JPHASE	determines if a specialty was defined to be included in subsegment 1 as a primary or alternate
2	IPHASE	determines if a specialty was defined to be included in subsegment 2, as a primary or alternate
3	OPT	determines if optional output reports are to be produced
4	IBITS	performs bit packing and unpacking for arrays that contain flow control codes
5	SET	sets binary representation for flow control codes
6	IPROB	determines if a specialty was defined to be a primary specialty in subsegment 1
7	VALID	determines if a specialty is an advanced entry specialty
8	SOPEN3	A subroutine in the UNIVAC EXEC 8 System Sort package. The amount of main storage, drum storage, and number of tape units available for use by the sort program is specified by SOPENx (x may have a value of 1, 2, 3, 4, or 5; successively greater amounts of data may be sorted by the higher value of x). SOPEN3 specifies 16,ØØØ words of main core storage, 2ØØ,ØØØ words of drum storage, and no tape units.
9	SRREL	A standard program linkage subroutine in the EXEC 8 System Sort package. This linkage releases a record to the Sort program.

TABLE III-1, Identification of Minor Supporting Functions or Subroutines Used in Matrix Generator Activity Flow Charts (concluded)

Identifying number	Function/ subroutine name	Description
10	SSORT	A standard program linkage subroutine in the EXEC 8 System Sort package. This linkage informs the Sort program that no more records are to be released. When this call is encountered, the sort is then performed.
11	SRRET	A standard program linkage subroutine in the EXEC 8 System Sort package. This linkage requests the sorted output records from the Sort program.

5. <u>Discussion and Flow Chart for FMPS Solution Activity</u>. - Since the FMPS is a proprietary program of UNIVAC, the name, functions, and calling sequence of programs is not available to the user. However, the general sequence of FMPS control language procedures is as illustrated in Figure III-16.*

---

*A detailed listing of these procedures is contained in paragraph 3f, Chapter IX, Catalogued Runstreams--the procedures are fully explained in FMPS Programers' Manual (reference 1).

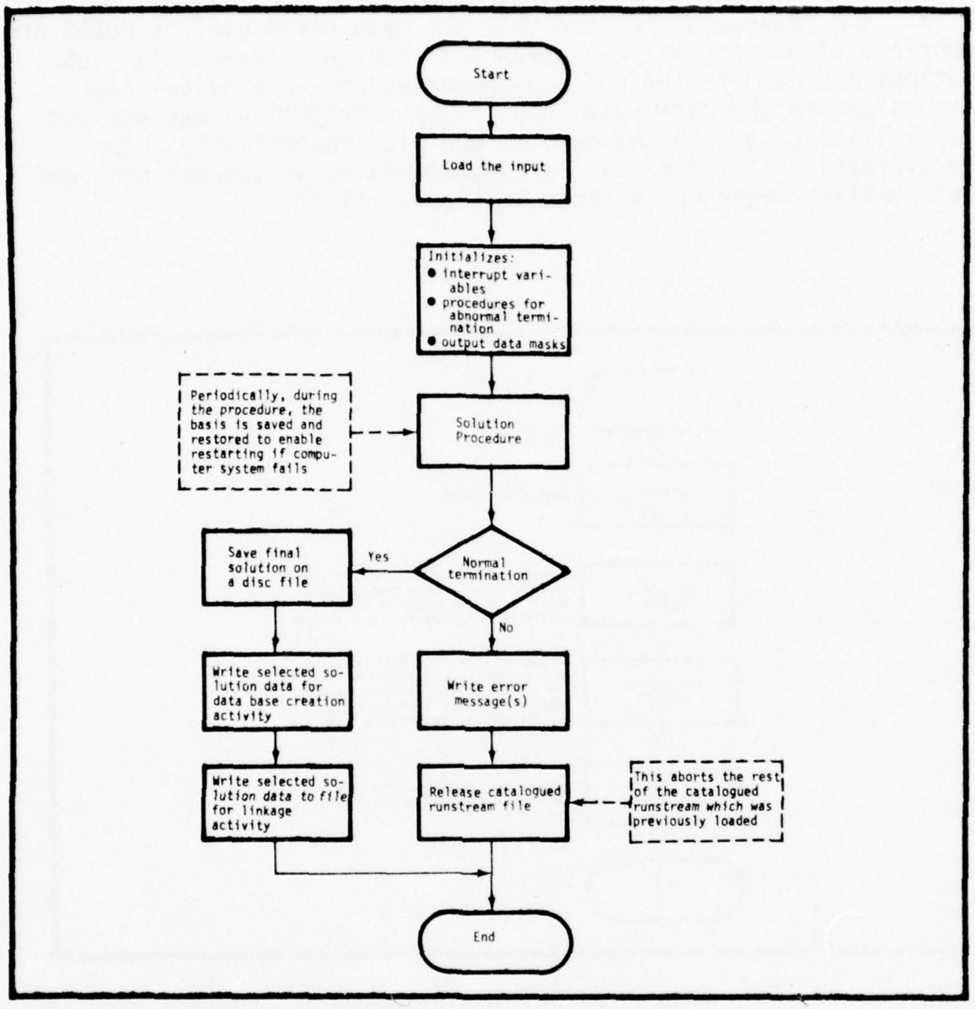


FIGURE III-16, General Sequence of FMPS Control Language Procedures

6. Discussion and Flow Charts for Data Base Creation Activity

a. The procedures for creating the data bases used in ODSAS are comprised of one executive program that controls three major subroutines and one program that re-sequences and updates selected data values in the data base input files. Program narratives and source listings are in Chapters XI and XII, respectively. The identification of the executive program and major subroutines, and their calling sequence is shown in Figure III-17.

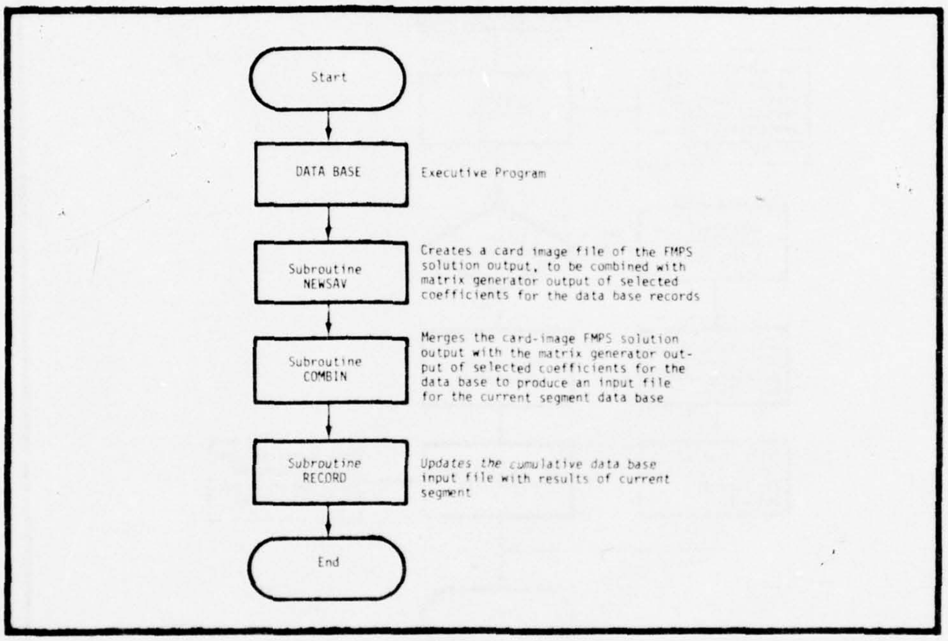


FIGURE III-17, Flow Chart for Calling Sequence of Programs Used in Data Base Creation Activity

b. The interaction of the first program (and associated subroutines) with the data files in creating the input files for input to the data bases is shown in Figure III-18.

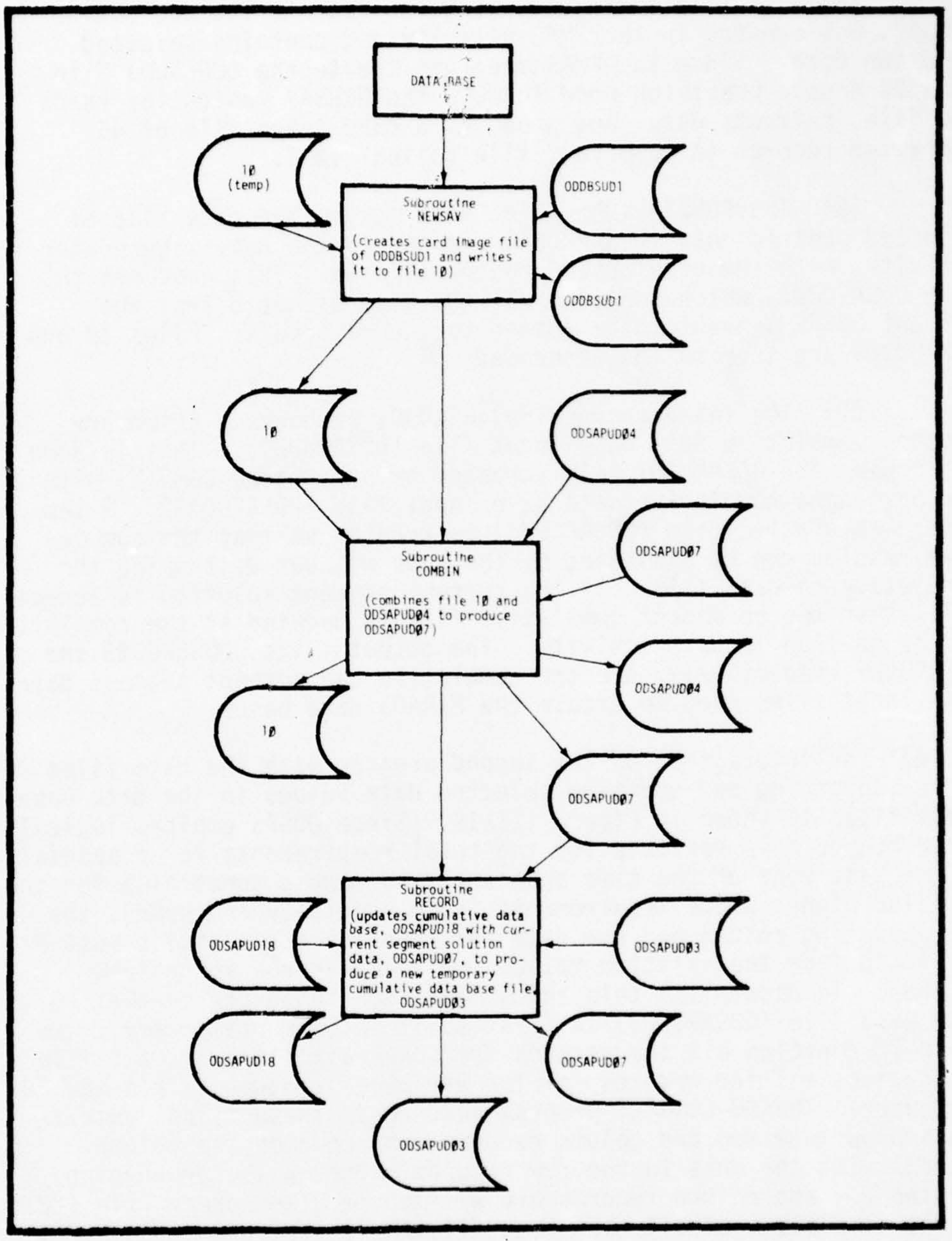


FIGURE III-18, Interaction of the Data Base Creation Programs With the Input and Output Files Needed to Create The Data Bases

(1) The file ODDBSUD1, which is input to the subroutine NEWSAV, was created in the FMPS activity and contains selected solution data. Since the FMPS programs create the ODDBSUD1 file in a 60 double precision word format, the NEWSAV subroutine reads the file, extracts data, and produces a card-image file of 45 character records (a temporary file called "1Ø").

(2) The COMBIN subroutine then merges the data file of selected coefficients (ODSAPUDØ4) created in the matrix generator activity, with the previously created file 1Ø. This produces the file ODSAPUDØ7, which contains all the data elements from the current ODSAS segment to be loaded to the data base. Files 1Ø and ODSAPUDØ4 are then no longer needed.

(3) The third subroutine, RECORD, produces a temporary updated cumulative data base input file (ODSAPUDØ3). This is done by merging the ODSAPUDØ7 file (created by subroutine COMBIN) with the permanent cumulative data base input file (ODSAPUD18). A temporary cumulative file (ODSAPUDØ3) is created so that the cumulative results can be evaluated by the user without destroying the cumulative-to-date file. If the current segment solution is acceptable, then the permanent cumulative file is updated at the completion of the on-line inquiry activity. The output files, ODSAPUDØ3 and ODSAPUDØ7 respectively, are the cumulative and current segment data base input files used to create the MIRADS data bases.

c. The interaction of the second program with the data files in re-sequencing and updating selected data values in the data base input files is shown in Figure III-19. Since ODSAS employs logical upper bounds on a variable for the total requirements for a specialty in the last year of the time span analyzed, and a constraint for the unfilled higher grade requirements (same specialty and year), the corresponding column and row data for these node capacities must be extracted from the solution values for selected row and column records. To accomplish this adjustment, the temporary cumulative data base file (ODSAPUDØ3) is first split into two temporary files (file 2Ø contains all the records for constraints (rows), and file 21 contains all the records for the variables (columns)) via the ED processor. The DB-CORRECT program then reads these files, selects the appropriate row and column records, and updates the column records with the data in the row records. During the processing, all the row and column records are written to a temporary file (22); at the completion of all updating, file 22 is copied to ODSAPUDØ3.

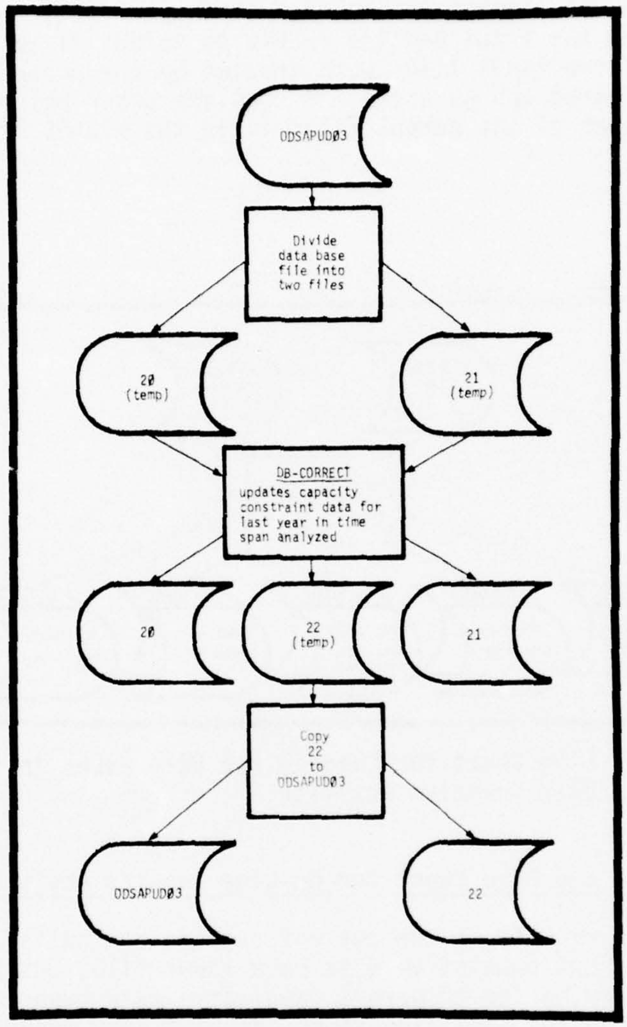


FIGURE III-19, Interaction of Data Base Creation Program With the Input and Output Files Needed to Re-sequence and Update Cumulative Data Base Input File

d. The second procedure in the data base creation activity is that of loading the data base. This is done with one source program, DBGEN (the source listing is included in Chapter XII). Figure III-20 depicts the input and the resulting output files from that program. The two input files were created by the procedures described in paragraph 6b above. A complete description of the function of each of the output files is in the MIRADS Implementation Manual.

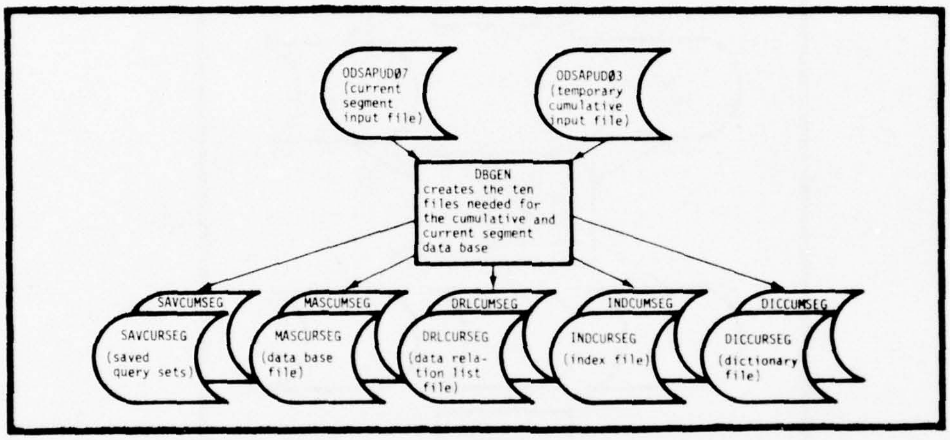


FIGURE III-20, Flow Chart for Loading the Data Bases in the Data Base Creation Activity

7. Discussion and Flow Chart for On-Line Inquiry Activity

a. If the results of the current segment are satisfactory, then the permanent cumulative data base input file, ODSAPUD18, is replaced by copying the temporary cumulative data base input file (ODSAPUDØ3) into the same file space occupied by ODSAPUD18 (via an EXEC 8 @COPY command). However, if the results are unsatisfactory, then a program to update selected files containing attrition and promotion rate and/or specialty requirements data values must be processed. The update program is a simple one; for each type change, the old parameter or rate value and the new parameter or rate value must be input, and if the old value matches what is already on file, the update is effected; otherwise the update is by-passed. The flow chart displaying the input that can be updated, the resulting output files, and the programed procedure is shown in Figure III-21.

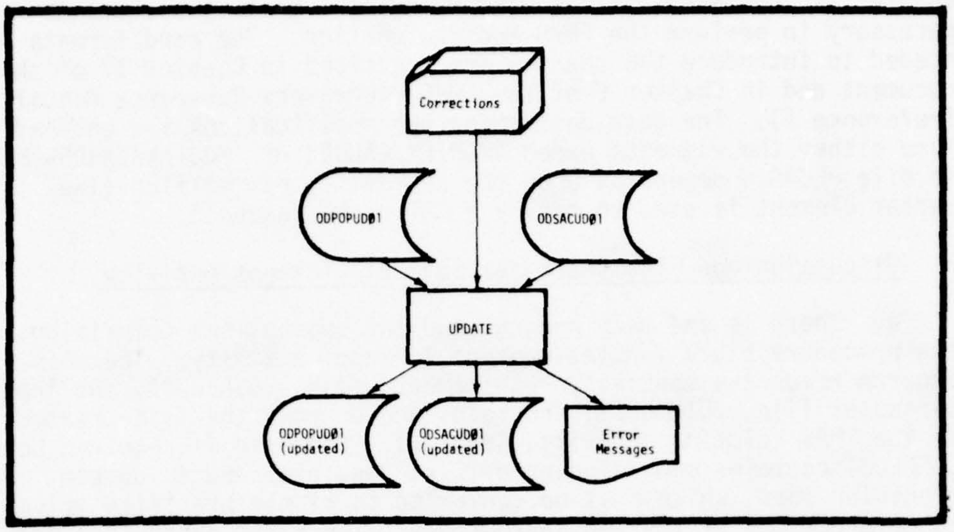


FIGURE III-21, Flow Chart of the UPDATE Program Procedures

b. If preferred specialty pairings, utilization ratios, or tour lengths need to be updated, then the procedure shown in Figure III-9 for creating or updating the specialty preferences file should be followed (by specifying UPDATE in the first parameter card, and then entering only the data values to be updated, in the same format as for file creation).

c. For certain types of changes there is an alternative procedure, (i.e., the FMPS Modify procedure), separate and distinct from those in a and b above, with which a new solution can be derived without again solving the entire problem.

(1) The types of changes that can be processed are:

(a) Upper and lower bounds

(b) Ranges for acceptable values of the b-coefficients (right-hand side coefficients) of the constraints

(c) Mathematical relationship by which the constraint is expressed (i.e., changing an equality to an inequality)

(d) Right-hand side coefficients

(e) Coefficients of variables

(2) A catalogued runstream, PFCAA.MODIFY, described in Chapter IX, contains all the EXEC 8 and FMPS control statements necessary to perform the FMPS Modify function. The card formats needed to introduce the changes are described in Chapter IV of this document and in Chapter 6 of the FMPS Programers Reference Manual (reference 1). The data describing the modifications are entered into either the elements named "MODIFY/CARDS" or "MODIFY/CARDS-CPT-LT" in file PFCAA., depending upon the segment to be modified (the latter element is used to modify a CPT or LT segment).

8. *Discussion and Flow Chart for Segment Linkages Activity*

a. There is one main program and two subroutines comprising the procedure block for the segment linkages activity. The main program reads the specialty requirements file, ODSACUDØ1, the input parameter file, ODINPUDØ1, and reads and decodes the file created in the FMPS solution activity, ODSOLUD1. Decoding is required because ODSOLUD1 contains many alphanumeric values, recorded in double precision form, which must be converted to single precision values to be processed by the two subroutines. The first subroutine produces the unfilled requirements report that displays the specialty requirements before and after updating with the current segment FMPS solution. The second subroutine takes the solution values and updates the specialty requirements file to reflect requirements filled-to-date and attrition in the system. Additional program narrative is provided in Chapter XI. Figure III-22 depicts the flow chart--with actual input and output file names--utilized and updated by the LINKAGE program:

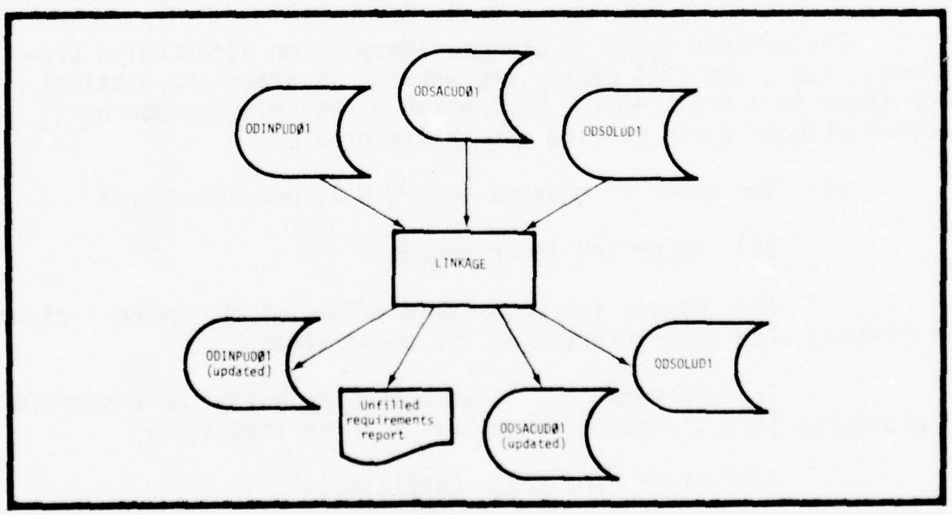


FIGURE III-22, Flow Chart of the Linkage Activity Procedures

b. As shown in the processing phase flow chart (Figure III-2), if the last grade segment has just completed, then the segment linkage activity is the final one to be run for that ODSAS processing series; otherwise, the remaining segments should be started by the user, and the input parameter file and the specialty requirements file are then updated as the first step in the new processing run. Following the update of the input parameter file and the specialty requirements file, the processing phase begins again with the matrix generator activity.

9. Separating Active and Inactive Records in the Cumulative Data Base

a. As described in paragraph 6 above, the cumulative data base is only updated to reflect the most recent changes to a particular grade's records, and records for grades in which all changes have been effected are put in a separate cumulative data base input file (e.g., when the LTC segment(s) are completed satisfactorily, COL records can no longer be updated since no COLs or LTCs promoted to COL will be identified in the remaining segments). This is done to save having to sort those records for each subsequent data base creation activity, even though the order would be unchanged and contents unaffected.

b. The flow chart for the procedure to separate the records is illustrated in Figure III-23. The separation is accomplished at the beginning of the MAJ, CPT, and LT processing phases, to separate COL, LTC, and MAJ records, respectively.

(1) The permanent cumulative data base file, ODSAPUD18, (which only contains records that can be updated), is input to the SEPARATE program. By comparing grade field in the data base records to the grade of the current segment, and determining if further updating is possible, the data base records are divided and written to two temporary files (16 and 17).

(2) File 17 contains those records that can no longer be updated. The ED processor is used to place those records in file 17 at the end of the cumulative input file of inactive records (ODCUMUDØ1).

(3) File 16 now contains only cumulative records that can be updated, and it is copied via EXEC 8 command to ODSAPUD18, the permanent cumulative file of records for the MIRADS data base.

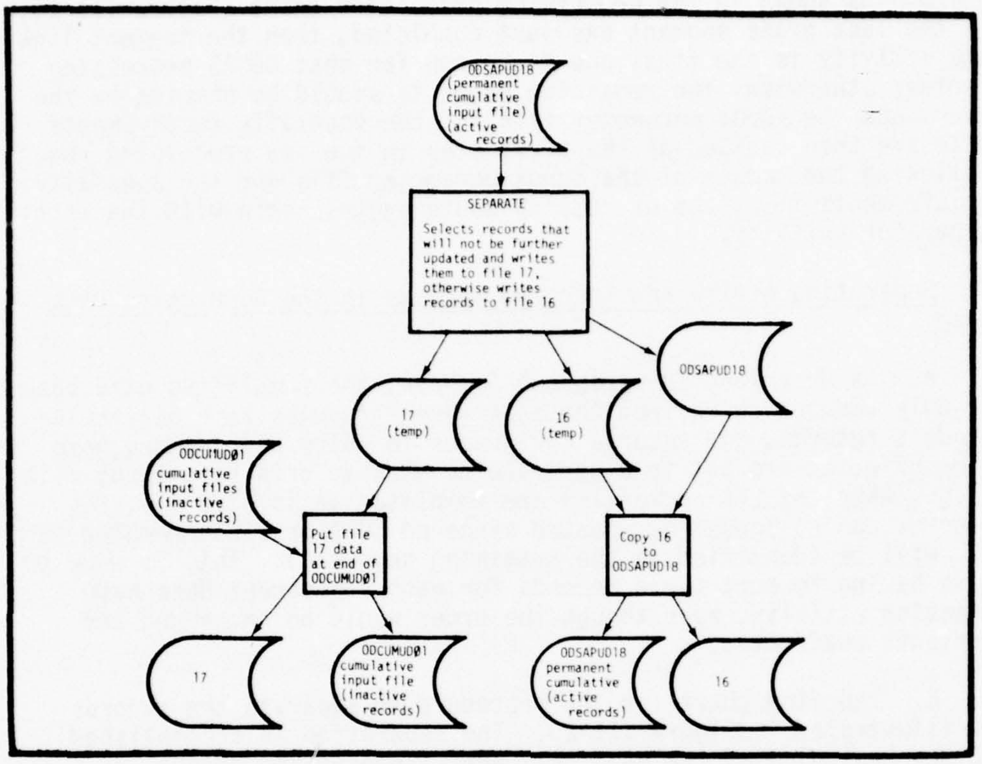


FIGURE III-23, Flow Chart of Procedures for Separating Active and Inactive Records in the Cumulative Data Base

10. Special Procedure for Accessing All Records in the Cumulative Data Base. - If the cumulative data base for all records (active and inactive) were needed for analysis, then the procedure described in paragraph 6c above would apply, except that ODCUMUDØ1 would be the only input file, and only the output files for the cumulative data base would be produced. Catalogued runstream PFCAA. IMPLEMENT/TOTAL (see Chapter IX) is used to accomplish this task.

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER IV
INPUT CARD FORMATS

1. Purpose. - The purpose of this chapter is to describe the formats and data items required for the card input to the ODSAS automated information system. The majority of the card input is processed in the initialization phase, and the remainder is applicable when updating or modifying data files in the processing phase (on-line inquiry activity).

2. Card Inputs for Initialization Phase. - As shown in Figure III-1, there are three card input files required for ODSAS; (1) the utilization ratio and tour lengths file, (2) the rates and populations file, and (3) the input parameters file. (The latter two files are combined into one input file, as explained in subparagraph b below.) The file names, card types and names, and data formats and descriptions are as follows:

a. Utilization Ratio and Tour Lengths File. - (Internal file name: ODTURUDØ1). - The data in this file are input in three different card types. The respective card types (assembled as shown in Figure IV-1) are described below:

(1) Card Type 1, Processing Mode Indicator. - One card only, punched as follows:

Card column	Format	Data description
1-6	A6	Indicator whether to create or update the file ("CREATE" or "UPDATE")

(2) Card Type 2, Prohibited Alternate Specialties. - As many cards as needed (normally only one needed), punched as follows:

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Number of specialties that cannot be alternate specialties, followed by a maximum of 1Ø OPMS specialty numbers identifying the prohibited alternate specialties

(3) Card Type 3, Preferred Specialty Pairs. - One card for each preferred specialty pair. Each OPMS specialty should appear in a type 3 card as a primary specialty at least once to ensure that a user-defined tour length is used for each specialty in each grade. A default value of 3 years is applied for the tour length if a specialty is not input as a primary specialty. A preferred specialty pair, "n/m," is generated by the program for every preferred specialty pair, "m/n," input via a card Type 3; therefore a card Type 3 is not required for the "n/m" pair, unless specification of a tour length (other than 3 years) for specialty "n" is desired. Type 3 cards may be assembled without regard to sequence, and are punched as follows:

Card column	Format	Data description
1-2	I2	primary specialty number
3-4	I2	alternate specialty number
5-9	F5.3	utilization ratio for primary/ alternate specialty pair in grade 0-6
10-11	I2	tour length of primary specialty in grade 0-6, multiplied by 1Ø
12-16	F5.3	utilization ratio for primary/ alternate specialty pair in grade 0-5
17-18	I2	tour length of primary specialty in grade 0-5, multiplied by 1Ø
19-23	F5.3	utilization ratio for primary/ alternate specialty pair in grade 0-4
24-25	I2	tour length of primary specialty in grade 0-4, multiplied by 1Ø

```
UTILIZATION AND TOUR LENGTHS FILE (ODTURUD01)  A/

                                                        CARD TYPES  A/
    PROCESSING MODE INDICATOR  A/

CREATE                                                  CARD 1

      PROHIBITED ALTERNATE SPECIALTIES  A/

4,11,12,13,14                                           CARD 2

    PREFERRED SPECIALTY PAIRS  A/

1115 .70030 .60030 .40030                               CARD 3
1128 .36030 .30030 .45030                               CARD 3
1131 .20030 .22030 .33030                               CARD 3
1135 .31030 .45030 .32030                               CARD 3
1137 .36030 .17030 .43030                               CARD 3
1141 .42030 .44030 .31030                               CARD 3
1142 .40030 .42030 .33030                               CARD 3
1145 .28030 .42030 .51030                               CARD 3

1197 .23030 .45030 .50030                               CARD 3
1215 .70030 .60030 .40030                               CARD 3
1220 .43030 .50030 .57030                               CARD 3
1231 .43030 .50030 .45030                               CARD 3

9586 .64030 .59030 .45030                               CARD 3
9587 .59030 .62030 .65030                               CARD 3
9588 .83030 .60030 .58030                               CARD 3
9591 .70030 .35030 .58030                               CARD 3
9592 .40030 .56030 .41030                               CARD 3
9795 .45030 .48030 .27030                               CARD 3

    A/ TITLES AND CARD TYPES INDICATED IN THIS FIGURE ARE FOR CON-
VENIENCE IN READING THE FIGURE, NOT INCLUDED IN ACTUAL INPUT.
```

FIGURE IV-1, Examples of Utilization Ratio and Tour Lengths Input Data Cards (File ODTURUDØ1)

b. Combined Input Parameters/Rates and Populations Files. - (Internal file name: ODR8SUDØ1). In the initialization phase, flow chart (Figure III-1) the Input Parameters file and the Rates and Populations file are shown as two separate entities. That distinction is made because of different types of data within those files. However, for processing, the data in those files are placed on a single input disc file, ODR8SUDØ1 (because of internal program logic considerations). The data in this file are input in 14 different card types (see examples in Figure IV-2). The first four card types contain input parameters. These are followed by seven card types (card types 5 through 11) containing rates and population data, and

three card types (card types 12 through 14) describing additional input parameters for the segmentation-within-grade option. The respective card types (assembled as shown in Figure IV-2) are described below:

(1) Card Type 1, Problem Size and Name. - A single card.

Card column	Format	Data description
1-2	I2	Number of OPMS specialties
3-4	I2	Number of years in the projection period
5-8	A4	First four characters of FMPS problem name (any four characters assigned by the user)
9-12	A4	Last four characters of FMPS problem name (any four characters assigned by the user)

(2) Card Type 2, Budget Authorizations by Grade. - One card only.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Total number of officers that are authorized by the budget for each grade at $T_Ø$, in ascending order, 2LT through COL.

(3) Card Type 3, Specialties to be Controlled. - As many cards as needed.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Number of primary specialties (up to 5Ø) that will have control of input constraints, followed by the appropriate OPMS specialty numbers (there must be at least one specialty specified). The number allowed as input for each specialty referenced in this card is specified in cards type 14.

(4) Card Type 4, Advanced Entry Specialties. - A single card.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Number of advanced entry specialties (AES), followed by up to 2Ø OPMS specialty numbers identifying the AES

Card types 5 through 11 are assembled in sets for successive officer grades, in descending order of grade (i.e., card types 5, 6, 7, and 9 for COL; followed by card types 5-9 for LTC, card types 5-9 for MAJ, card types 5-9 for CPT, card types 5-11 for 1LT, and card types 5-11 for 2LT).

(5) Card Type 5, Year-of-Service Spread. - One card for each grade.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Officer grade (e.g. 6), followed by the least number of YOS, and the greatest number of YOS to be represented in that grade

(6) Card Type 6, Population by YOS. - As many cards as needed for each grade.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	The number of officers in the grade indicated in card type 2, in each YOS represented in that grade. Entries must be in ascending order of YOS.

(7) Card Type 7, Attrition (Including Promotion) Rates by YOS. - As many cards as needed for each grade.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Annual attrition rates (that include promotion as a form of attrition) for the grade indicated in card type 5, for each YOS represented in that grade, beginning with the lowest YOS. Rates are percentages

		expressed as decimals (e.g., 15.95 percent per year is expressed as .1595; 2Ø percent is expressed as .2Ø). Not more than nine entries per grade for 1LT and 2LT, because of system design considerations; there are no limits on other grades.

(8) Card Type 8, Attrition (Excluding Promotion) Rates by YOS. - As many cards as needed for each grade (normally only 1 needed). Note: This card type does not apply for grade 6, COL.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Annual attrition rates (that do not include promotion as a form of attrition) for the grade indicated in card type 5, for each YOS represented in that grade, beginning with the lowest YOS. Rates are percentages expressed as decimals (e.g., 7.62 percent per year is expressed as .Ø762; 1Ø percent is expressed as .1Ø).

(9) Card Type 9, Overfill and Minimum Fill. - One card only.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Percentages, expressed as decimals, for overfilling requirements in each year, and specifying minimum fill percentages for each specialty in each year. Four decimal values are required for each node (except Grade 6, for which only two values are required), as follows: 1) percentage of overfill of requirements for grade in card Type 5 2) minimum percentage of fill of requirements for grade in card Type 5 3) percentage of overfill of requirements for next higher grade than one specified in card Type 5 (not required for Grade 6) 4) minimum percentage of fill of requirements for next higher grade

than one specified in card Type 5 (not required for Grade 6)

(10) Card Type 1Ø, Additional LT Attrition (Including Promotion) Rates. - One card for the 1LT set and one card for the 2LT set only. This card type does not apply to other grades.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Nine annual attrition rates (that include promotion as a form of attrition) applicable to lieutenants with service beyond the last YOS specified in card Type 5

(11) Card Type 11, Additional LT Attrition (Excluding Promotion) Rates. - One card for the 1LT set and one card for the 2LT set only. This card type does not apply to the other grades.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Nine annual attrition rates (that do not include promotion as a form of attrition), applicable to lieutenants with service beyond the last YOS specified in card Type 5

(12) Card Type 12, Segmentation Indicator. - A single card.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Grade number (followed by a Ø or 1, for grades 6, 5, and 4). A "Ø" indicates that no segmentation within grade is desired; a "1" indicates that segmentation within grade is desired.

(13) Card Type 13, Segment 1 Maximum Fill Percentages by Specialty. - As many cards as needed. This card type must be present, even though segmentation within grade is not desired.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Each OPMS specialty number, followed by a decimal (from Ø.Ø to 1.Ø) indicating maximum amount of a specialty's requirements

fillable in segment 1. (1.Ø is equivalent to a maximum of 1ØØ percent fill in segment 1. Enter 1.Ø if no restrictions are desired).

(14) Card Type 14, Upper Limits for Controlled Specialties. - As many cards as needed. An upper limit must be specified for every specialty (see default rule below for entry for a specialty that is not identified in card Type 3).

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Each OPMS specialty number, followed by a real number (to one decimal place) representing the maximum number of officers identified with the specialty number (either as a primary or alternate specialty), that can enter the network; if no upper bound applies, enter a default value of 1.Ø.

Note: A set of card types 13 and 14 is required for each of grades 6, 5, and 4, regardless whether the grade is to be segmented. A default value of 1.Ø should be entered for each specialty for each unsegmented field grade. The sets are entered in descending order of grade (i.e., card types 13 and 14 for Grade 6, followed by card types 13 and 14 for Grade 5, and ending with card types 13 and 14 for Grade 4).

3. Card Inputs for Processing Phase. - As explained in Chapter III (Paragraph 2 c (2) (b) 4), if either the solution or the cumulative results at the completion of any segment are unacceptable to the user, then the user may change policies, parameters, or requirements data values. Procedures for implementing such changes are described in Chapter VI (Paragraph 7). The data changes are introduced via card input in two categories: (1) to update the specialty requirements file and/or the attrition and promotion rates file and thereafter solve the LP problem from the beginning, (2) to modify an existing FMPS solution so that the LP problem solution can be recomputed without a complete rerun.

```
          COMBINED INPUT PARAMETERS/RATES AND POPULATIONS FILE (ODR8SUD01) A/

                                                                    CARD TYPE A/
                        PROBLEM SIZE AND NAME A/
4505TESTFLOW                                                        CARD 1

                    BUDGET AUTHORIZATIONS BY GRADE A/
0,17695,24552,12031,8514,3335                                       CARD 2

                     SPECIALTIES TO BE CONTROLLED A/
12,11,12,13,14,21,31,44,53,71,86,95,97                              CARD 3

                      ADVANCED ENTRY SPECIALTIES A/
16,15,41,45,46,47,48,49,51,52,53,54,86,91,93,95,97                  CARD 4
--------------------------------------------------------------------------------
                            (COLONEL SET) A/

                       YEAR-OF-SERVICE SPREAD A/
6,18,30                                                             CARD 5

                          POPULATION BY YOS A/
6,21,70,132,241,510,464,577,465,343,283,164,59                      CARD 6

         ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.139,.079,.116,.062,.060,.100,.154,.195,.228,.211,.348,.322,.517    CARD 7

                   OVER-FILL AND MINIMUM FILL A/
.1,.7                                                               CARD 9
--------------------------------------------------------------------------------
                    (LIEUTENANT COLONEL SET) A/

                       YEAR-OF-SERVICE SPREAD A/
5,14,28                                                             CARD 5

                          POPULATION BY YOS A/
42,142,282,945,1564,1380,1483,1048,792,311,211,136,80,60,38         CARD 6

         ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.049,.030,.028,.041,.050,.099,.299,.592,.300,.284,.296,.247         CARD 7
.300,.706,.357                                                      CARD 7

         ATTRITION (EXCLUDING PROMOTION) RATES BY YOS A/
.049,.030,.023,.033,.035,.073,.264,.243,.230,.227,.196              CARD 8
.300,.706,.357                                                      CARD 8

                   OVER-FILL AND MINIMUM FILL A/
.1,.7,.1,.7                                                         CARD 9
--------------------------------------------------------------------------------
                             (MAJOR SET) A/

                       YEAR-OF-SERVICE SPREAD A/
4,9,21                                                              CARD 5

                          POPULATION BY YOS A/
0,40,343,2300,2156,2161,2275,1821,929,0,0,0,0                       CARD 6

         ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.047,.051,.033,.030,.070,.091,.177,.683,.406,.321,.221,.679,.688    CARD 7

         ATTRITION (EXCLUDING PROMOTION) RATES BY YOS A/
.047,.034,.029,.023,.024,.026,.090,.041,.090,.078,.188,.670,.688    CARD 8

                   OVER-FILL AND MINIMUM FILL A/
.1,.7,.1,.7                                                         CARD 9
```

FIGURE IV-2, Examples of Combined Input Parameters/Rates and Populations Data Cards (File ODR8SUDØ1) (continued on next page)

```
(CAPTAIN SET) A/
YEAR-OF-SERVICE SPREAD A/
3,4,14                                                                 CARD 5
POPULATION BY YOS A/
0,2750,3818,4032,3361,3613,4274,2704,0,0,0                             CARD 6
ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.333,.082,.079,.056,.062,.051,.448,.333,.200,.192,.25                  CARD 7
ATTRITION (EXCLUDING PROMOTION) RATES BY YOS A/
.333,.082,.078,.046,.038,.028,.054,.065,.074,.154,.063                 CARD 8
OVER-FILL AND MINIMUM FILL A/
.1,.7,.1,.7                                                            CARD 9

(FIRST LIEUTENANT SET) A/
YEAR-OF-SERVICE SPREAD A/
2,1,6                                                                  CARD 5
POPULATION BY YOS A/
0,0,4099,3400,521,80                                                   CARD 6
ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.100,.597,.283,.937,.741,.571                                          CARD 7
ATTRITION (EXCLUDING PROMOTION) RATES BY YOS A/
.100,.597,.283,.000,.741,.571                                          CARD 8
OVER-FILL AND MINIMUM FILL A/
.1,.7,.1,.7                                                            CARD 9
ADDITIONAL LT ATTRITION (INCLUDING PROMOTION) RATES A/
.8,.99,.99,.99,.99,.99,.99,.99,.99                                     CARD 10
ADDITIONAL LT ATTRITION (EXCLUDING PROMOTION) RATES A/
.8,.99,.99,.99,.99,.99,.99,.99,.99                                     CARD 11

(SECOND LIEUTENANT SET) A/
YEAR-OF-SERVICE SPREAD A/
1,1,6                                                                  CARD 5
POPULATION BY YOS A/
5366,4780,0,0,0,0                                                      CARD 6
ATTRITION (INCLUDING PROMOTION) RATES BY YOS A/
.082,.942,.984,.769,.833,.999                                          CARD 7
ATTRITION (EXCLUDING PROMOTION) RATES BY YOS A/
.082,.000,.984,.769,.833,.999                                          CARD 8
OVER-FILL AND MINIMUM FILL A/
.1,.7,.1,.7                                                            CARD 9
ADDITIONAL LT ATTRITION (INCLUDING PROMOTION) RATES A/
.8,.99,.99,.99,.99,.99                                                 CARD 10
ADDITIONAL LT ATTRITION (EXCLUDING PROMOTION) RATES A/
.8,.99,.99,.99,.99,.99                                                 CARD 11
```

FIGURE IV-2, Examples of Combined Input Parameters/Rates and Populations Data Cards (File ODR8SUDØ1) (continued on next page)

```
                    SEGMENTATION INDICATOR A/
8,1,5,1,4,1                                                              CARD 12

          SEGMENT 1 MAXIMUM FILL PERCENTAGES, BY SPECIALTY A/
11,1.0,12,1.0,13,1.0,14,1.0,15,1.0,21,1.0,25,1.0,26,1.0,27,1.0           CARD 13
28,1.0,31,1.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,43,1.0,44,1.0           CARD 13
45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0,54,1.0           CARD 13
71,1.0,72,1.0,73,1.0,74,1.0,75,1.0,76,1.0,77,1.0,81,1.0,82,1.0           CARD 13
83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0           CARD 13

             UPPER LIMITS FOR CONTROLLED SPECIALTIES A/
11,1000.0,12,300.0,13,500.0,14,200.0,15,1.0,21,350.0,25,1.0,26,1.0       CARD 14
27,1.0,28,1.0,31,100.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,13,1.0         CARD 14
44,50.0,45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0          CARD 14
54,1.0,71,1.0,72,1.0,73,1.0,74,1.0,75,1.0,76,1.0,77,1.0,81,1.0,82,1.0    CARD 14
83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0           CARD 14
--------------------------------------------------------------------------------
                       (GRADE 5 (LTC)) A/

          SEGMENT 1 MAXIMUM FILL PERCENTAGES, BY SPECIALTY A/
11,1.0,12,1.0,13,1.0,14,1.0,15,1.0,21,1.0,25,1.0,26,1.0,27,1.0           CARD 13
28,1.0,31,1.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,43,1.0,44,1.0           CARD 13
45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0,54,1.0           CARD 13
71,1.0,72,1.0,73,1.0,74,1.0,75,1.0,76,1.0,77,1.0,81,1.0,82,1.0           CARD 13
83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0           CARD 13

             UPPER LIMITS FOR CONTROLLED SPECIALTIES A/
11,2000.0,12,750.0,13,1500.0,14,750.0,15,1.0,21,800.0,25,1.0,26,1.0      CARD 14
27,1.0,28,1.0,31,400.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,43,1.0         CARD 14
44,150.0,45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0         CARD 14
82,1.0,83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0    CARD 14
--------------------------------------------------------------------------------
                       (GRADE 4 (MAJ)) A/

          SEGMENT 1 MAXIMUM FILL PERCENTAGES, BY SPECIALTY A/
11,1.0,12,1.0,13,1.0,14,1.0,15,1.0,21,1.0,25,1.0,26,1.0,27,1.0           CARD 13
28,1.0,31,1.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,43,1.0,44,1.0           CARD 13
45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0,54,1.0           CARD 13
71,1.0,72,1.0,73,1.0,74,1.0,75,1.0,76,1.0,77,1.0,81,1.0,82,1.0           CARD 13
83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0           CARD 13

             UPPER LIMITS FOR CONTROLLED SPECIALTIES A/
11,2000.0,12,1000.0,13,2000.0,14,1000.0,15,1.0,21,1000.0,25,1.0,26,1.0   CARD 14
27,1.0,28,1.0,31,850.0,35,1.0,36,1.0,37,1.0,41,1.0,42,1.0,43,1.0         CARD 14
44,250.0,45,1.0,46,1.0,47,1.0,48,1.0,49,1.0,51,1.0,52,1.0,53,1.0         CARD 14
54,1.0,71,1.0,72,1.0,73,1.0,74,1.0,75,1.0,76,1.0,77,1.0,81,1.0           CARD 14
82,1.0,83,1.0,86,1.0,87,1.0,88,1.0,91,1.0,92,1.0,93,1.0,95,1.0,97,1.0    CARD 14

A/ TITLES, CARD NAMES AND CARD TYPES INDICATED IN THIS FIGURE ARE FOR CON-
VENIENCE IN READING THE FIGURE. THESE DATA ARE NOT INCLUDED IN ACTUAL INPUT.
```

FIGURE IV-2, Examples of Combined Input Parameters/Rates and Populations Data Cards (File ODR8SUDØ1) (concluded)

a. Updating Specialty Requirements and/or Attrition and Promotion Rates File (Internal file names: ODSACUDØ1 and ODPOPUDØ1, respectively). - Data to update these files are input in selected combinations from among eight types of cards and placed in the "UPDATE/CARDS" element of the "PFODSAP" program file via the ED processor. The respective card types (assembled as shown in the examples in Figure IV-3) are described as follows:

(1) Card Type 1, File Update Codes. - One card only, always used.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Parameter indicating the file(s) to be updated. Code values are: 1 - Specialty Requirements file 2 - Attrition and Promotion Rates file 3 - Both files

(2) Card Type 2, Requirements Change. - Required when card type 1 specifies 1 or 3. As many cards as needed, one requirements change per card.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Year number (1-1Ø, representing $T_Ø$ - T_9, respectively), grade number, OPMS specialty number, old requirements value, new requirements value

(3) Card Type 3, Rate Type Change. - One card for each grade/type of rate change to be made. Required when card Type 1 specifies 2 or 3. Each card Type 3 must be followed by either a card Type 4, or a card Type 5, or card types 6 and 7 (always paired), or a card Type 8, as applicable.

Card column	Format	Data description
1	I1	Grade number
2-7 (left justified)	A6	Type of rate to be updated: ATTHI - Attrition rate applicable to promotees ATTLO - Attrition rate applicable to those remaining in grade

PRMT - Promotion rate

OFLOLO - Percentage of overfill allowed for requirements in a specified grade

UFLOLO - Percentage of underfill allowed for requirements in a specified grade

OFLOHI - Percentage of overfill allowed for requirements one grade higher than the specified grade

UFLOHI - Percentage of underfill allowed for requirements one grade higher than the specified grade

AES - Advanced entry specialties

CPTREM - Percentages of captains with less than 8 YOS

(4) Card Type 4, Attrition/Promotion Rate Change. - Required when card Type 3 specifies ATTHI, ATTLO, or PRMT.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	Nine weighted average attrition rates of the appropriate type (as specified in card Type 3)

(5) Card Type 5, Overfill and Minimum Fill Rate Change. - Required when card Type 3 specifies OFLOLO, UFLOLO, OFLOHI, or UFLOHI.

Card column	Format	Data description
1-8Ø	Free (input values separated by commas)	One decimal value to replace the overfill or minimum fill parameter specified in card Type 3

(6) Card Type 6, Total Number of AES. - Required when card Type 3 specifies AES. A card Type 6 must be followed immediately by a card Type 7.

Card column	Format	Data description
1-80	Free (input values separated by commas)	Total number of advanced entry specialties that are to be identified on card Type 7

(7) Card Type 7, AES Identification Numbers. - Required when card Type 3 specifies AES. A card Type 7 must follow immediately after a card Type 6.

Card column	Format	Data description
1-80	Free (input values separated by commas)	Advanced entry specialty numbers

(8) Card Type 8, Rates for CPT Population With Less Than 8 YOS. - Required when card Type 3 specifies CPTREM.

Card column	Format	Data description
1-80	Free (input values separated by commas)	Nine decimal rates representing the proportion of the total captain population with less than 8 YOS at any node in the network for each of the nine years (the maximum number of years that can be considered in the model). Rates must be in descending numerical order, or input will be rejected

b. Modification of an Existing FMPS Solution

(1) The data (in the formats described below) to modify an existing FMPS solution is placed in the element MODIFY/CARDS or MODIFY/CARDS-CPT-LT of file "PFCAA" via the ED processor. These two elements are described and illustrated in Chapter IX, Paragraph 4. The procedure to modify an existing FMPS solution is described in Chapter VI, Paragraph 7.

```
          UPDATE OF SPECIALTY REQUIREMENTS FILE (ODSACUD01)   A/
        AND/OR ATTRITION AND PROMOTION RATES FILE (ODPOPUD01)

                                                          CARD TYPE A/
          FILE UPDATE CODES  A/
3                                                            CARD 1

         REQUIREMENTS CHANGE  A/
3,4,21,109,94                                                CARD 2
2,5,11,46,59                                                 CARD 2

     ------------------------------------------------------------
           RATE TYPE CHANGE  A/
6ATTLO                                                       CARD 3

      ATTRITION/PROMOTION RATE CHANGE  A/
.036,.037,.036,.036,.036,.038,.040,.040,.040                 CARD 4

     ------------------------------------------------------------
           RATE TYPE CHANGE  A/
4OFLOLO                                                      CARD 3

   OVER-FILL AND MINIMUM FILL RATE CHANGE  A/
.12                                                          CARD 5

     ------------------------------------------------------------
           RATE TYPE CHANGE  A/
3AES                                                         CARD 3

         TOTAL NUMBER OF AES  A/
18                                                           CARD 6

       AES IDENTIFICATION NUMBERS  A/
41,45,46,47,48,49,51,52,53,54,86,91,93,95,97                 CARD 7

     ------------------------------------------------------------
           RATE TYPE CHANGE  A/
3CPTREM                                                      CARD 3

 RATES FOR CPT POPULATION WITH LESS THAN 8 YOS  A/
.75,.64,.53,.42,.31,.20,.15,.10,.05                          CARD 8

    A/  TITLES, CARD NAMES AND CARD TYPES INDICATED IN THIS FIGURE ARE FOR CON-
VENIENCE IN READING THE FIGURE. THESE DATA ARE NOT INCLUDED IN ACTUAL INPUT.
```

FIGURE IV-3, Examples of Cards to Update Specialty Requirements File (ODSACUDØ1) and/or Attrition and Promotion Rates File (ODPOPUDØ1)

(2) Each data value to be modified in an existing FMPS solution requires a single card. Use as many cards as needed (not to exceed 5ØØ). Card format is as follows:

Card column	Format	Data description
2-3	A2 (left justify)	Type of constraint or bound.

		1) FMPS type of constraint (i.e., L, E, or G; where L indicates $\leq$, E indicates =, G indicates $\geq$).
		2) Type of bound on a variable (i.e., LO or UP; where LO indicates lower bound and UP indicates upper bound)
		3) Leave blank when changing a variable's coefficient, a right-hand side coefficient, or a range value.
5-12	2A4 (left justify)	Constraint, Chapter, or Variable Name.
		1) Constraint name (see Appendix D), if modifying the FMPS type of constraint
		2) FMPS chapter name (i.e., .RANGES. or .BOUNDS.), if modifying a range or a bound value
		3) Variable's name (see Appendix D), if modifying a variable's coefficient
		4) The right-hand side chapter's name (i.e., RHS), if modifying a b-coefficient (right-hand side)
15-22	2A4 (left justify)	Row or Column Name (see Appendix D).
		1) Row name, except when modifying a bound upon a variable
		2) Column name when modifying a bound upon a variable
		3) Leave blank when modifying the FMPS type of constraint

25-36	F12.4	New coefficient for a variable, or new value for a range, bound or b-coefficient. Leave blank when modifying the FMPS type of constraint

Note: Chapter 6, Paragraph 6.2.3, of the FMPS programers reference manual, provides complete details on the types of modifications that can be effected.

(3) Examples of cards to modify data values in an existing FMPS solution are shown in Figure IV-4.

```
     MODIFICATION OF DATA VALUES  A/
     IN AN EXISTING FMPS SOLUTION

E    N135TREQ
UP   .BOUNDS.     X0013              96.0000
LO   .BOUNDS.     W01321             16.0000
     .RANGES.     N127TREQ           25.0000
     RHS          UR9597              2.3000
     W04591       R09145              0.5000

------------
  A/  TITLE NOT INCLUDED IN INPUT.
```

FIGURE IV-4, Examples of Cards to Modify Data Values in an Existing FMPS Solution

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER V
ODSAS PRINTED OUTPUT

1. Purpose. - The ODSAS information system produces both standard and optional output reports. In addition, reporting of as-required information to the user is available through an on-line inquiry capability. The purpose of this chapter is to display examples of system outputs that are available for printing if the user desires them for analysis or retention as hard-copy. The reports are produced in what is called a "Breakpoint" file. Through this procedure, the report is written directly to a computer disc file rather than to a printer. In this way, the user can look at the report via a cathode ray tube (CRT) terminal and evaluate the need to print the report. Sample outputs are arranged according to the order in which they are produced, starting with the Initialization Phase and ending with the linkage activity of the Processing Phase.

2. Initialization Phase Reports. - In the Initialization Phase, there are two types of standard output reports. These serve to inform the user on data which have been derived for use as input to the Processing Phase, thus permitting verification of the data before use.

a. Positional Requirements Reports. - The first type (identified in Figure III-1 as Output 6) contains the calculated positional requirements by specialty and by grade. Figure V-1 is a sample of the report on requirements by specialty; Figure V-2 is a sample of the report on requirements by grade. These reports are derived from data contained in the Personnel Structure and Composition System (PERSACS).

b. Attrition and Promotion Rates Reports. - The second report type contains the derivation and computations used to produce the input rates to the system. An excerpt from that report for LTC is in Figure V-3. Numbers boxed-in are examples of the rates input to the matrix generator.

				TOTAL REQUIREMENTS FOR SPECIALTY 49	
YEAR OF SERVICE	GRADE 0-2	GRADE 0-3	GRADE 0-4	GRADE 0-5	GRADE 0-6
T0	10	60	175	220	43
T1	11	82	190	225	47
T2	9	83	189	225	47
T3	9	83	193	225	47
T4	9	83	188	225	47
T5	9	83	189	225	47
T6	9	83	183	225	47
T7	9	83	189	225	47
T8	9	83	189	225	47
T9	9	83	189	225	47
TOTALS	93	806	1875	2245	466

FIGURE V-1, Sample PERSACS Requirements Report - by Specialty (for Specialty 49)

TOTAL REQUIREMENTS FOR GRADE 6

SPECIALTY	YEAR T-0	YEAR T-1	YEAR T-2	YEAR T-3	YEAR T-4	YEAR T-5	YEAR T-6	YEAR T-7	YEAR T-8
11	250	264	263	262	262	262	262	262	262
12	118	115	115	115	115	115	115	115	115
13	186	184	183	183	183	183	183	193	183
14	68	74	75	75	75	73	73	73	73
15	4	4	4	4	4	4	4	4	4
21	137	136	139	137	137	137	137	137	137
25	44	41	41	41	41	41	41	41	
26	33	33	33	33	33	33			
27	34	32	32	32					
28	4	3						6	6
31						11	11	11	11
					20	20	20	20	20
			10	10	10	10	10	10	10
	9	9	8	3	3		8	8	8
91	78	80	80	73	73	73	79	73	79
92	117	112	113	111	111	111	111	111	111
93	38	36	35	35	35	35	35	35	35
95	37	46	46	44	44	44	44	44	44
97	82	77	78	77	77	77	77	77	77
98	310	323	325	321	321	321	321	321	321
TOTALS	3315	3333	3316	3279	3279	3277	3277	3277	3277

FIGURE V-2, Sample PERSACS Requirements Report - by Grade (for Grade 6)

****OFFICERS GRADE-5 AT T-0****

YOS	STARTING POPULATION	RATE W/O PROM	RATE W/PROM	PROMOTIONS (POPULATION*RATE DIFFERENCE)	POPULATION LESS PRMT	ATTRITION IN GRADE
13-14	42.0	.0490	.0490	.00	42.00	2.06
14-15	142.0	.0300	.0300	.00	142.00	4.26
15-16	282.0	.0230	.0280	1.41	280.59	6.45
16-17	945.0	.0330	.0410	7.56	937.44	30.94
17-18	1564.0	.0350	.0500	23.46	1540.54	53.92
18-19	1380.0	.0730	.0990	35.88	1344.12	98.12
19-20	1483.0	.2640	.2990	51.91	1431.10	377.81
20-21	1048.0	.2340	.5920	375.18	672.82	157.44
21-22	792.0	.2300	.3000	55.44	736.56	169.41
22-23	311.0	.2270	.2840	17.73	293.27	66.57
23-24	211.0	.2650	.2960	6.54	204.46	54.18
24-25	136.0	.1960	.2470	6.94	129.06	25.30
25-26	80.0	.3000	.3000	.00	80.00	24.00
26-27	60.0	.7060	.7060	.00	60.00	42.36
27-28	38.0	.3570	.3570	.00	38.00	13.57
TOTAL POPULATION	8514.0		TOTAL PROMOTIONS	582.04	TOTALS 7931.96	1126.38

ATTRITION RATE IN GRADE 1126.38/ 7931.96 = $\boxed{.1420}$

PROMOTION RATE 582.04/ 8514.00 = $\boxed{.0684}$

FIGURE V-3, Excerpt of Attrition and Promotion Rates Computation Report

3. Processing Phase Reports

a. Matrix Generator Activity. - There is one standard report and one optional report produced in the matrix generator activity. The standard report is a statistical summary, and the optional report contains the internal programing codes used in the matrix generator to record the paths in the networks.

(1) Statistical Summary. - This standard report, printed in two parts, contains the key requirements data and characteristics of the problem to be solved.

(a) The first part (Figure V-4) shows both the unfilled higher grade requirements (passed down from the preceding segment) and the requirements for the grade of the current segment. (Requirements data values may be greater than actual computed requirements if the user opts to provide input directing that requirements may be overfilled by a percentage of the authorized value.) Requirements values in this summary report are used as the capacities of the nodes. Column 2 of the report (entitled PCT AUTH) contains the maximum percentage fill allowed for a specialty in that segment. The values in Column 2 would be less than 1ØØ.ØØ only if the user specifies a lower percentage and opts to segment the processing within grade.

(b) The second part of the statistical summary (Figure V-5) shows the problem size, in terms of the total number of rows for each constraint type, and a summation of all constraints (rows). This total number of rows should match the matrix statistics produced by FMPS described in subparagraph b below. Additionally, the report displays key parameter values (number of specialties, number of years in the projection period, and number of preferences) applicable to the current segment.

(2) Optional Report. - This report, consisting of three parts, displays certain coded information generated and used internally by the matrix generator program. The three parts--Restrictive Flow Codes for the grade being processed, Restrictive Flow Codes for the higher (promotee) grade, and Last Exit Year Points for the grade being processed--are described in subparagraphs (a) (b) and (c), respectively.

(a) Part 1, Restrictive Flow Codes for the Grade Being Processed. - This part appears as a square matrix comprised of a row and a corresponding column for each OPMS specialty. The matrix contains coded information concerning the existence and nature of flows between the respective specialties (nodes) for officers remaining in grade. A separate matrix is printed for each year in

LTC UNSEGMENTED ODSAS DATE 031676 PAGE 6

UNFILLED REQUIREMENTS FOR GRADE O-6

	PCT AUTH	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
SPECIALTY 11	100.00	0	74	103	156	199	228				
SPECIALTY 12	100.00	0	30	40	60	78	90				
SPECIALTY 13	100.00	0	45	65	92	115	131				
SPECIALTY 14	100.00	0	19	26	41	49	55				
SPECIALTY 15	100.00	0	1	1	2	3	3				
SPECIALTY 21	100.00	0	59	88	132	165	189				
SPECIALTY 25	100.00	0	11	18	26	31	34				
SPECIALTY 26	100.00	0	11	14	22	30	35				
SPECIALTY 27	100.00	0	12	15	23						
SPECIALTY 28	100.00	0									
SPECIALTY 31											
						30	35				
				1	2	2	2				

LTC UNSEGMENTED ODSAS DATE 031676 PAGE 8

REQUIREMENTS FOR GRADE O-5

	PCT AUTH	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
SPECIALTY 11	100.00	689	676	678	676	665	643				
SPECIALTY 12	100.00	331	332	331	331	327	321				
SPECIALTY 13	100.00	559	553	554	543	530	514				
SPECIALTY 14	100.00	289	271	255	252	250	246				
SPECIALTY 15	100.00	36	36	36	36	36	36				
SPECIALTY 21	100.00	499	503	502	500	498	496				
SPECIALTY 25	100.00	194	190	189	189	188	186				

FIGURE V-4, ODSAS Statistical Summary Report--Part 1

```
COLONEL UNSEGMENTED

  SEGMENT NUMBER **0** FOR GRADE EQUAL TO **6**


                    CONSTRAINTS BY TYPE
                OBJECTIVE FUNCTION                  1
                FLOW CONSERVATION                 315
                REQUIREMENTS - TOTAL              225   (HIGHER GRADE =        0)
                CONTROL OF X-ARCS                3158
                CONTROL OF Y-ARCS(PROMOTEES)        0
                CONTROL OF INPUT                    8
                KEY ARC RELATIONSHIPS             316
                   TOTAL CONSTRAINTS             4023


                NUMBER OF SPECIALTIES EQUAL        45
                NUMBER OF YEARS EQUAL               5
                NUMBER OF PREFERENCES EQUAL       632
```

FIGURE V-5, ODSAS Statistical Summary Report--Part 2

the timespan under consideration. A sample Grade 6 matrix for 45 specialties is shown at Figure V-6. OPMS specialties are listed on the left in ascending numerical order; specialty numbers are not printed out for column headings because of print space limitations. The code value (Ø,1,2, or 3) at row/column intersections indicates the following:

1. Code Value Ø. - No flow between the two specialties (nodes) during that year. Note that, at year zero, values of Ø appear only where the row and column are for the same specialty; 3's, described later, are substituted at year zero. Values of Ø appear in matrices for subsequent years. A code of Ø has meaning only for naming a restrictive flow constraint with a prefix "R" and then only when the two specialty numbers are different. When the specialty numbers are the same, a restrictive flow constraint of the N_ _ _LINC type is defined.

2. Code Value 1. - There is an equality flow control constraint upon the flow between the two specialties during that year.

```
RESTRICTIVE FLOW CODES FOR GRADE  6 AT YEAR    0
CODES:  0 - NO FLOW  1 - FLOW(= CONSTRAINT) 2 -DELETE CODE 3 - UPPER BOUND

SPECIALTY 11   0 3 3 3 1 3 3 3 3 1 1 1 3 1 1 1 3 3 1 1 1 1 1 1 3 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 3 2
SPECIALTY 12   3 0 3 3 1 3 3 3 3 1 1 1 3 3 1 3 3 3 1 1 1 1 1 1 3 1 1 1 3 3 3 3   1 3 3 3 3 3 3 1 1 3 3 2
SPECIALTY 13   3 3 0 3 1 3 3 3 3 1 1 3 1 1 3 3 3 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 2
SPECIALTY 14   3 3 3 0 1 3 3 3 3 3 1 3 3 1 1 3 3 1 1 1 1 1 1 1 1 1 3 1 3 3 3   3       3 3 1 3   2
SPECIALTY 15   1 1 1 1 0 3 3 3 3 1 1 3 1 1 1 3 3 1 1 1 1 1 1 3 1 1 3 1 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 2
SPECIALTY 21   3 3 3 3 3 0 3 3 3 3 3 1 3 3 1 1 3 3 1 3 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 2
SPECIALTY 25   3 3 3 3 3 3 0 1 1 1 3 3 3 1 1 1 3 3 1 1 1 1 1 1 3 1 1 3 1 3 3 3 3     3 3   3 3 3 1 3 3 2
SPECIALTY 26   3 3 3 3 3 3 1 0 1 1 3 3 3 1 1 1 3 3 1 3 1 1 1 1 3 1 1 3 1 3 3 3 3 3 3 3   3 3 3 3 1 3 3 2
SPECIALTY 27   3 3 3 3 3 3 1 1 0 3 3 3 3 1 3 3 3 3 3 3 1 3 3 1 3 1 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2
SPECIALTY 28   1 1 3 3 3 3 1 1 3 0 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3   3 3
SPECIALTY 31   1   1 3 1 3 3 3 3 3 0 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3   3 3 3     3 1 3 2 3
SPECIALTY 35   1 1 1 1 1 1 3 3 3 3 1 0 1 1 1   3 3 1 3 1 1 1 1 3 1 2 3 3 3 3 3 3 3     3   3 3   3 3
SPECIALTY 36   3 3 3 3 3   3 3 3 3 1 1 0 1 1 3 3 3 1 3 1 1 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3   3 3 3 3 3
SPECIALTY 37   1 3 1 3 1 3 1 1 1 3 3 1 1 0 1 1 3 3 1 3 1 1 1 1 1 3 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3
SPECIALTY 41   1 1 1 1 1 1 1 1 3 3 3 1 1 1 0 1 3 3 1 1 3 3 3 3 1 1 3 1 3   1 1 3 3 3 3 3 1 1 1 1 1 3 2 3
SPECIALTY 42   1 3 3 1 1 1 1 1 3 3 3 1 3 1 1 0 1 3 1 1 1 3 3 3 3 1 3 3 3 3 3 1 3 3 3 3 3 1 1 1 1 1 3 2 3
SPECIALTY 43   3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 2 3 3 3
SPECIALTY 44   3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 0 1 3 1 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3   3
SPECIALTY 45   1 1 1 1 1 1 1 1 3 3 3 1 1 1 1 1 3 1 0 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 1 3 3 1 1 3 2 3
SPECIALTY 46   1 1 1 1 1 3 1 3 3 1 3 3 3 3 1 2 3 3 3 0 3 3 3 3 3 3 3 3   3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
SPECIALTY 47   1 1 1 1 1 1 1 1 1 3 3 1 1 1 3 1 3 1 3 3 0 3 3 3 3 3 3 3 3 3 1 1 3 1 3 3 3 3 3   1 1 3 2 3
SPECIALTY 48   1 1 1 1 1 1 1 1 3 3 3 1 1 1 3 3 3   3 3 3 0 3 3 3   3 3 3 3 1 1 3 1 3 3 3 3 3   1 1 3 2 3
SPECIALTY 49   1 1 1 1 1 1 1 1 3 3 3 1 3 1 3 3 3 3 3 3 3 0 3 3 3 3 1 3 1 1 1 3 1 3 3 3 3 3 3 1 1 3 2 3

SPECIALTY 88   3 3 3 3 3 3 3 3 3 3 3 3 3   1 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 1 3 0 1 1 3 2 3
SPECIALTY 91   1 1 1 3 1 1 3 3 3 3 3 3 3 3 1 1 1 3 1 3 1 1 1 1 3 1 1 1 3 1 1 1 1 1 3 3 3 1 1 1 0 1 1 1 2
SPECIALTY 92   1 1 1 1 1 1 1 1 3 3 1 3 3 3 1 1 1 3 1 3 1 1 1 1 3 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 0 1   2
SPECIALTY 93   1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3   3 3 1 1 1 3 3 3 1 2 0 3 3
SPECIALTY 95   3 3 3 3 3 3 3 3 3 3 1 3 3 3 1 1 3 3 1 3 1 1 1 1 3 1 3 1 3 3 3 3 3 3 3 3 1 1 1 1 1 3 0 2
SPECIALTY 97   1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 3 3 3 1 1 3 2 0

RESTRICTIVE FLOW CODES FOR GRADE  6 AT YEAR    1
CODES:  0 - NO FLOW  1 - FLOW(= CONSTRAINT) 2 -DELETE CODE 3 - UPPER BOUND

SPECIALTY 11   0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0   0 0 0 0 0   1 1 1 0 2
SPECIALTY 12   0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 2
SPECIALTY 13   0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 2
SPECIALTY 14   0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 2
```

FIGURE V-6, Sample of Matrix Generator Activity Optional Report, Part 1 - Restrictive Flow Codes for Grade n at Year y

3. Code Value 2. - Same as a code value of 1, but the equality flow control constraint is redundant and is therefore deleted from consideration as an explicit constraint to save processing time. The constraint upon the flow in the output arc from a specialty to the highest numbered preferred specialty was arbitrarily chosen as the one to be deleted for each specialty, for each year, $T_Ø$ through $T_{(NYRS-1)}$. The constraint is redundant because the flow in this output arc is uniquely determined: that flow is the amount of flow remaining after the flow conservation constraint upon a node and the flow control constraints on all the other output arcs have been satisfied. For example, (see Figure V-7), the flow conservation constraint on node 14 at T_3 specifies that the node input equals the node output or,

$$X_{21414} + X_{29714} = X_{31414} + X_{31497}$$

A restrictive flow constraint (N_ _ _LINC type) specifies that:

$$X_{31414} = Ø.25\ X_{21414} + Ø.5Ø\ X_{29714}$$

Therefore, if the input to node 14 were as shown in Figure V-7 (i.e., $X_{21414} + X_{29714} = 1ØØ$), then the output must equal 1ØØ (because of the flow conservation constraint). The restrictive flow constraint upon X_{31414} yields a value of 3Ø for X_{31414} (i.e., (Ø.25 x 8Ø) + (Ø.5 x 2Ø) = 3Ø). The remainder of the output, 1ØØ-3Ø = 7Ø, can go to only one place (i.e., X_{31497}); thus a constraint is not needed.

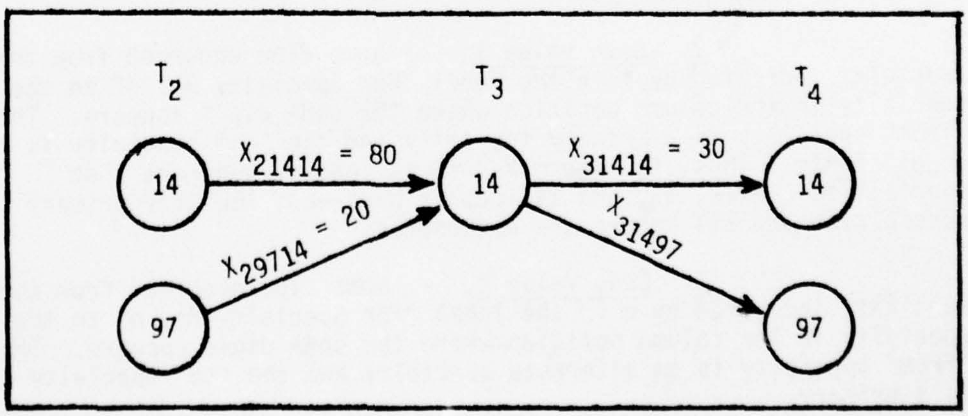


FIGURE V-7, Example of Redundant Flow Control Constraint

4. Code Value 3. - This is a special form of "no flow," substituted for a code value of Ø in the year Ø matrix only. The digit 3 serves as a program flag to omit unnecessary steps during processing. If an entire column in the year Ø matrix contains only 3's (except that a Ø always appears where the row and the column are for the same specialty), there are no flows into that specialty from any others, i.e., that specialty is not included in any preferred specialty pair in the matrix. Consequently there can be no flow out of that node for the next year, i.e., the upper bound is set equal to zero.

(b) Part 2, Restricted Flow Codes for the Higher (Promotee) Grade. - This part appears as a square matrix comprised of a row and a corresponding column for each OPMS specialty. The matrix contains coded information concerning the flows between the respective specialties (nodes) for officers of the grade population being processed who are promoted into the next higher grade during the timespan under consideration. A separate matrix is printed for each terminal year in the timespan (e.g., no matrix is printed for the year $T_Ø$; the first matrix, labeled year T_1, contains information on flows beginning at $T_Ø$ and ending at T_1; the second, labeled year T_2, is for flows beginning at T_1 and ending at T_2, and so on). A sample 45 x 45 matrix for the LTC segment, containing flow information on LTCs promoted to COL, is shown at Figure V-8. The term "For Specialty No. n" in Figure V-8 indicates the "from" specialty. Each such term is followed by a line of coded single digit entries, each digit position representing a "to" specialty corresponding to the OPMS specialties, in ascending order from left to right, respectively. The single digit code value entries (Ø, 1, 2, 3, or 4) indicate the following:

1. Code Value Ø. - No flow between the two specialties during that year.

2. Code Value 1. - Some flow occurred from the specialty indicated by n in the label "For Specialty No. n" to the specialty in the column position where the code digit appears. The "from" specialty is a primary specialty and the "to" specialty is an alternate. Thus, in Figure V-8 it is readily apparent that specialties 11, 12, 13, and 14--combat arms--are the only primary specialties and all others are alternates.

3. Code Value 2. - Some flow occurred from the specialty indicated by n in the label "For Specialty No. n" to the specialty in the column position where the code digit appears. The "from" specialty is an alternate specialty and the "to" specialty is a primary.

```
RESTRICTED FLOW CODES FOR GRADE  6  YEAR T  1
CODES: 0=NO FLOW  1=FLOW FROM PRIMARY 2=FLOW FROM ALTERNATE  3=FLOW FROM BOTH  4=PERTAINS TO LT SEG

FOR SPECIALTY NO. 11
 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1
FOR SPECIALTY NO. 12
 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 13
 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 14
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
FOR SPECIALTY NO. 15
 2 2 2 2 0 0 0 0 0 0 2 2 0 2 2 2 0 0 2 2 2 2 2 2 0 2 2 0 2 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2
FOR SPECIALTY NO. 21
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 2 0 0 2 0 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2
FOR SPECIALTY NO. 25
 0 0 0 0 0 0 0 2 2 2 0 0 0 2 2 2 0 0 2 2 2 2 2 2 0 2 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2

RESTRICTED FLOW CODES FOR GRADE  6  YEAR T  2
CODES: 0=NO FLOW  1=FLOW FROM PRIMARY 2=FLOW FROM ALTERNATE  3=FLOW FROM BOTH  4=PERTAINS TO LT SEG

FOR SPECIALTY NO. 11
 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1
FOR SPECIALTY NO. 12
 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 13
 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 14
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
FOR SPECIALTY NO. 15
 2 2 2 2 0 0 0 0 0 0 2 2 0 2 2 2 0 0 2 2 2 2 2 2 0 2 2 0 2 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2
FOR SPECIALTY NO. 21
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 2 0 0 2 0 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2
FOR SPECIALTY NO. 25
 0 0 0 0 0 0 0 2 2 2 0 0 0 2 2 2 0 0 2 2 2 2 2 2 0 2 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2

RESTRICTED FLOW CODES FOR GRADE  6  YEAR T  4
CODES: 0=NO FLOW  1=FLOW FROM PRIMARY 2=FLOW FROM ALTERNATE  3=FLOW FROM BOTH  4=PERTAINS TO LT SEG

FOR SPECIALTY NO. 11
 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1
FOR SPECIALTY NO. 12
 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 13
 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
FOR SPECIALTY NO. 14
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
FOR SPECIALTY NO. 15
 2 2 2 2 0 0 0 0 0 0 1 3 0 3 3 1 0 0 1 3 1 3 3 3 0 1 3 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 3
FOR SPECIALTY NO. 21
 0 0 0 0 0 0 0 0 0 0 0 3 0 0 3 3 0 0 3 0 1 3 1 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 3
FOR SPECIALTY NO. 25
 0 0 0 0 0 0 0 3 1 3 0 0 0 1 3 3 0 0 1 1 1 3 1 3 0 1 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 3
```

FIGURE V-8, Sample of Matrix Generator Activity Optional Report, Part 2 - Restricted Flow Codes for the Higher (Promotee) Grade

4. Code Value 3. - Some flow occurred into the "to" specialty where the "from" specialty is both a primary and an alternate. Such flows occur because of officer reassignments from their primary to their alternate specialty at different years in the projection period. Thus, a reassignment of some officers from their primary to an alternate specialty at T_3 occurs when officers with opposite primary and alternate specialties are reassigned.

5. Code Value 4. - This value appears only in a matrix for the LT segment. It indicates that the flow in the Y-arc contains flow from an earlier alternate specialty designation point, and that designations also occur in the interval in which the arc appears.

(c) Last Exit Year Points for Grade Being Processed. - This part also appears as a square matrix with rows and columns corresponding to the OPMS specialties, but is displayed only once for each grade. A sample grade 6 matrix for 45 specialties is shown at Figure V-9. The positive (i.e., nonzero) values in the matrix indicate the number of increments in which the members of the initial ($T_\emptyset$) population depart a specialty. For example, in the first line of Figure V-9, the value 4 which appears at the intersection of Specialty 11 and Specialty 15 (the fifth column), indicates that the officer(s) from among those who entered the network as part of the grade 6 population with specialty pairing 11/15 at $T_\emptyset$, departed their assignment in specialty 11 in four increments ($T_\emptyset$, T_1, T_2, and T_3). (All departures are functions of tour length and utilization ratios.)

b. FMPS Activity. - Most of the printed output from the FMPS activity consists of diagnostic messages concerning FMPS internal logic at periodic intervals during processing, and is explained in the FMPS documentation (Reference 1). The two outputs of primary concern are the matrix statistics and the detailed listing of the solution.

(1) A sample matrix statistics output is at Figure V-10. The number of rows should be the same as shown in the statistical summary report from the matrix generator activity. The number of columns is the number of arcs in the network. There will always be a value of "1" for RHS, indicating one right-hand side to the ODSAS equations. The density value refers to the percentage of nonzero elements in the LP matrix (.17 is Ø.17 percent). Max-COL-nz's is the maximum number of nonzero elements in any column. Elements (Figure V-10) include the total number of (nonzero and zero) elements in the LP matrix. The largest and smallest element values are then shown, followed by major and minor errors encountered on input--these last two values should be zero.

LAST EXIT YEAR POINTS FOR GRADE 6

NOTE: POSITIVE VALUES INDICATE LAST YEAR WHEN T-0 POPULATION DEPARTS A SPECIALTY

SPECIALTY																																													
11	0	0	0	0	4	0	0	0	0	2	2	2	0	2	4	4	0	0	2	2	2	2	4	2	0	2	4	4	0	0	0	0		0	0	0	0	0	0	0	2	2	2	0	2
12	0	0	0	0	4	0	0	0	0	4	4	2	0	0	2	0	0	0	2	2	2	2	4	2	0	2	2	4	0	0	0	0	0	4	0	0	0	0	0	0	2	2	0	0	4
13	0	0	0	0	4	0	0	0	0	0	4	2	0	4	2	0	0	0	2	2	2	2	2	2	2	2	2	4	0	0	0	0	0	0	0	0	0	0	0	0	4	2	0	0	2
14	0	0	0	0	4	0	0	0	0	0	0	4	0	0	4	2	0	0	2	2	2	2	4	2	2	4	2	2	0	2	0	0	0	0	0	0	0	0	0	0	0	4	0	0	4
15	2	2	2	2	0	0	0	0	0	0	2	2	0	2	4	4	0	0	2	2	2	2	4	2	0	2	4	0	2	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	2
21	0	0	0	0	0	0	0	0	0	0	0	4	0	0	4	4	0	0	4	0	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4
25	0	0	0	0	0	0	0	2	2	2	0	0	0	2	2	2	0	0	2	2	2	2	2	2	0	2	2	0	2	0	0	0	0	0	0	0	0	0	0		0	2	0	0	2
26	0	0	0	0	0	0	4	0	2	4	0	0	0	2	4	4	0	0	4	0	2	4	4	2	0	2	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	2
27	0	0	0	0	0	0	4	4	0	0	0	0	0	2	0	0	0	0	0	0	4	0	0	4	0	4	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
28	4	4	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	4	4	4	0	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	2	0
35	4	4	4	4	4	2	0	0	0	0	4	0	4	2	4	4	0	0	4	0	4	2	4	2	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	2	2	0	2	2	0	0	0	2	0	2	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	4	0	4	0	4	0	4	4	4	0	0	4	4	0	4	4	0	0	4	0	2	4	4	4	0	4	4	4	0	0	4	0	0	0	0	0	0	0	0	0	0	0		0	0
41	4	2	2	4	4	2	2	4	0	0	0	2	4	2	0	4	0	0	2	2	0	0	0	0	4	2	0	2	0	0	4	2	0	0	0	0	0	2	2	4	2	4	0	2	0
42	4	0	0	2	4	2	4	2	0	0	0	2	0	2	4	0	4	0	2	2	2	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	4	2	4	2	4	0	2	0
51	4	4	4	4	4	2	4	2	2	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	0	4	2	4	4	0	0	4	0	0	0	2	4	0	2	0
52	0	0	4	4	0	2	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0
53	4	4	4	4	4	2	4	2	2	0	0	4	4	4	4	4	0	4	0	0	0	0	0	0	0	0	0	4	2	0	4	4	0	4	0	0	0	0	2	0	4	4	0	4	0
54	4	2	2	2	4	2	2	2	2	0	0	4	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	4	4	0	0	0	0
71	4	4	4	4	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	4	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	2	2
72	0	0	0	0	4	0	4	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0		0	0	0	0	0	0	4	0	0	4
73	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	4	4	0	0	4
74	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	4	0	2	4	4	2	2	4	4	0	0	0	0	2	0	0	0	0	0	0	0	0	4	4	0	0	2
75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	4	0	2	2	4	2	2	4	4	0	0	2	2	0	0	0	0	0	0	0	0	0	4	4	0	0	2
76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	0	0	0	2
77	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	4	0	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0	4	4	0	0	4
81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	4	2	0	2
82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	4
83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	2	0	4
86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	2	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	4	4	4	4	0	2	0
87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0		0	0	0	0	0	0	2	0	0	0		0	0	0	0	0	0	0	4	0	0	2	2	0	4	0
88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	2	2	0	2	0
91	4	2	4	0	4	2	0	0	0	0	0	0	0	0	4	4	4	0	2	0	4	2	4	2	0	4	4	4	0	4	4	4	4	4	0	0	0	4	2	4	0	4	4	2	4
92	4	4	4	4	4	2	4	4	0	0	2	0	0	0	4	4	4	0	4	0	4	2	4	2	0	4	0	4	2	2	4	2	0	2	4	4	4	4	2	4	4	0	2	4	4
93	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	4	0	0	0	4	4	0	0	0
95	0	0	0	0	0	0	0	0	0	0	4	0	0	0	4	4	0	0	4	0	4	2	4	4	0	4	0	4	0	0	0	0	0	0	0		0	4	4	4	4	4	0	0	4
97	4	4	2	4	4	2	4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	2	2	2	4	4	2	4	4	0	0	0	4	4	0	4	0

Figure V-9, Sample of Matrix Generator Activity Optional Report, Part 3 - Last Exit Year Points for Grade n

```
COLONEL UNSEGMENTED

    1 ..  1     TITLE  ODSAS FMPS COL - MAJ
    2 ..  2     CALL ENTER(LP,DUBLFMPS)
    3 ..        CALL INPUT

NAME            TESTFLOW

BUFFER SIZES (WORDS) ARE.. MATRIX =  4256 INVERSE =  3584

MATRIX STATISTICS
ROWS.........   4023
COLUMNS......   4105
RHS..........      1
DENSITY......    .17
MAX-COL-NZ'S     549
ELEMENTS.... 27537
LARGEST.....   .333500+004
SMALLEST....   .228255-001
MAJOR ERRORS       0
MINOR ERRORS       0
```

FIGURE V-10, Sample Matrix Statistics Output from FMPS

(2) The detailed listing of the solution is normally written out to a computer disc file for retention, possible future printing, and user inquiry via a text editor. The solution is written in three sections: identifier, rows, and columns.

(a) A sample identifier section is shown in Figure V-11. The figure reflects that the LP problem had an optimal status, the objective function (OBJECTIV) had a maximum value of 1814.444443, and 3Ø43 interactions were required to solve the problem.

```
COLONEL UNSEGMENTED

            ODSAS FMPS COL - MAJ

IDENTIFIER SECTION

          PROBLEM... NAME..
                     MODE.. LP
                     CLASS. LP
                     STATUS OPTIMAL.
          FUNCTIONAL NAME.. OBJECTIV
                     OBJECT MAXIMIZE
                     VALUE.     1814.444443
          RESTRAINT  NAME.. B-VECTOR
          ITERATION  COUNT.      3043
```

FIGURE V-11, FMPS Solution Output--Identifier Section

(b) An excerpt of the rows section is at Figure V-12. All the rows have unique names relating to constraint types. The naming convention is described in Appendix D.

(c) The rows of primary interest are those for node capacity, i.e., CREQ and TREQ, since all others except control of input constraints are specified as equalities to zero. For instance, flow conservation constraints require that the node input, less the node output, be equal to zero. If the row name ends in TREQ, then the capacity is for the current grade plus the unfilled higher grade. If the name ends in CREQ, then the capacity is for only the unfilled higher grade requirements. For each row dealing with node capacities, the activity is the number of officers assigned to a specialty for a given year. The slack activity is the difference between the activity and the upper limit (node capacity) shown in Column 5. The lower limit, Column 6, would appear as a zero or "None" unless minimum level of flow into a node has been specified. Such a minimum is used in the CPT segment to require filling of at least the unfilled higher grade requirements. The data values in the three rightmost columns are used for postoptimality analysis.

(d) The columns section (Figure V-13) is similar to the rows section in that there are corresponding columns for name, activity, and lower and upper limits. For columns however, activity value represents the amount of flow (dual-qualified officers) in a path segment described by the arc name. The input cost is the coefficient of a variable in the objective function. A value other that zero would appear only for the variables representing the arcs exiting the network. The variables have unique names, XN_{nn}, where N indicates the last year and nn is the specialty or node number. The lower and upper limit columns contain the minimum and maximum values, respectively, that the flow in the arcs may attain. The reduced cost values are used in postoptimality analysis.

(e) The types of constraints and variables--and thus the composition of the rows and columns sections--differ according to the grade segment being processed. A summary of the constraints and variables included in each ODSAS grade segment is shown in Table V-1.*

---

*See Chapter II of the ODSAS Study Report (Reference 4) for a complete description of the application of the constraint types and variables. See also Appendix D, this documentation, for naming conventions.

COLONEL UNSEGMENTED

ODSAS FMPS COL - MAJ

SECTION 1 - ROWS PRIMAL-DUAL OUTPUT

NUMBER	..NAME..	AT	...ACTIVITY...	SLACK ACTIVITY	.LOWER LIMIT	.UPPER LIMIT	.DUAL ACTIVITY	..INPUT COST..	.REDUCED COST.
1	OBJECTIV	FR	1814.444443	-1814.444443	NONE	NONE	-1.000000	.000000	-1.000000
2	W011G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
3	W012G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
4	W013G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
5	W014G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
6	W015G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
7	W021G070	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
110	N125TREQ	UL	45.000000	.000000	NONE	45.00000	.000000	.000000	.000000
111	N025G070	EQ	.000000	.000000	.00000	.00000	-.236415	.000000	-.236415
112	N025LINC	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
113	N126TREQ	UL	47.000000	.000000	NONE	47.00000	.000000	.000000	.000000
114	N026G070	EQ	.000000	.000000	.00000	.00000	-.236415	.000000	-.236415
115	N026LINC	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
116	N127TREQ	UL	46.000000	.000000	NONE	46.00000	.000000	.000000	.000000
117	N027G070	EQ	.000000	.000000	.00000	.00000	-.236415	.000000	-.236415
118	N027LINC	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
119	N128TREQ	UL	3.000000	.000000	NONE	3.00000	-.000000	.000000	.000000
120	N028G070	EQ	.000000	.000000	.00000	.00000	-.236415	.000000	-.236415
121	N028LINC	EQ	.000000	.000000	.00000	.00000	-.000000	.000000	.000000
122	N131TREQ	UL	49.000000	.000000	NONE	49.00000	.000000	.000000	.000000
123	N031G070	EQ	.000000	.000000	.00000	.00000	-.236415	.000000	-.236415
124	N031LINC	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
125	N135TREQ	UL	100.000000	.000000	NONE	100.00000	.000000	.000000	.000000
1085	UR9195	UL	4.100000	.000000	-4.10000	4.10000	.000000	.000000	.000000
1086	UR9197	BS	.000000	4.850000	-4.85000	4.85000	.000000	.000000	.000000
1087	UR9293	UL	4.925000	.000000	-4.92500	4.92500	.000000	.000000	.000000
1088	UR9295	BS	.000000	5.425000	-5.42500	5.42500	.000000	.000000	.000000
1089	UR9297	LL	-6.175000	12.350000	-6.17500	6.17500	.000000	.000000	.000000
1090	UR9597	BS	.000000	5.100000	-5.10000	5.10000	.000000	.000000	.000000
1091	R01115	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
1092	R01121	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
1093	R01128	EQ	.000000	.000000	.00000	.00000	.000000	.000000	.000000
1094	R01131	EQ	.000000	.000000	.00000	.00000	-.000000	.000000	.000000

FIGURE V-12, FMPS Solution Output--Rows Section

COLONEL UNSEGMENTED

ODSAS FMPS COL - MAJ

SECTION 2 - COLUMNS PRIMAL-DUAL OUTPUT

NUMBER	..NAME..	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
4024	X00011	BS	558.177971	.000000	.000000	689.000000	.000000
4025	X00012	BS	178.000000	.000000	.000000	178.000000	.000000
4026	X00013	BS	225.756887	.000000	.000000	308.000000	.000000
4027	X00014	BS	91.281246	.000000	.000000	118.000000	.000000
4028	X00015	BS	8.300000	.000000	.000000	NONE	.000000
4029	X00021	BS	96.000000	.000000	.000000	96.000000	.000000
4030	X00025	BS	56.224984	.000000	.000000	NONE	.000000
4031	X00026	BS	48.452777	.000000	.000000	NONE	.000000
4032	X00027	BS	14.147396	.000000	.000000	NONE	.000000
4033	X00028	BS	1.612500	.000000	.000000	NONE	.000000
4597	W01591	BS	.000000	.000000	.000000	NONE	.000000
4598	W02191	BS	.000000	.000000	.000000	NONE	.000000
4599	W04191	BS	.000000	.000000	.000000	NONE	.000000
4600	W04291	BS	.000000	.000000	.000000	NONE	.000000
4601	W04391	**	-.000003	.000000	.000000	NONE	.000000
4602	W04591	BS	4.343748	.000000	.000000	NONE	.000000
4603	W04791	BS	2.312500	.000000	.000000	NONE	.000000
4790	X19797	BS	62.208576	.000000	.000000	NONE	.000000
4791	X21111	BS	154.832436	.000000	.000000	NONE	.000000
4792	X21212	BS	54.599618	.000000	.000000	NONE	.000000
4793	X21313	BS	85.941478	.000000	.000000	NONE	.000000
4794	X21414	BS	35.699173	.000000	.000000	NONE	.000000
4795	X21515	BS	1.895991	.000000	.000000	NONE	.000000
4796	X22121	BS	134.786160	.000000	.000000	NONE	.000000
4797	X22525	BS	15.647215	.000000	.000000	NONE	.000000
4798	X22626	BS	21.896703	.000000	.000000	NONE	.000000
4799	X22727	BS	27.301240	.000000	.000000	NONE	.000000
4800	X22828	BS	1.562078	.000000	.000000	NONE	.000000
8084	XN11	BS	70.631643	1.000000	.000000	305.000000	.000000
8085	XN12	BS	28.605311	1.000000	.000000	122.000000	.000000
8086	XN13	BS	43.106357	1.000000	.000000	178.000000	.000000
8087	XN14	BS	16.476624	1.000000	.000000	73.000000	.000000

FIGURE V-13, FMPS Solution Output--Columns Section

TABLE V-1, Summary of Application of Variables, Constraints and Promotions in the ODSAS Grade Segments

Segment	Variables			Constraints [a]						Promotions
				Node Capacity		Flow Control				
	W arc	X arc	Y arc	TREQ	CREQ	LINC	CINC	R_ _ _	RES_ _ _	
COL	Yes	Yes	No [b]	Yes	No	Yes	No	Yes	No	Yes [b]
LTC and MAJ	Yes	Yes	Yes [c]	Yes	Yes	Yes	Yes	Yes	Yes	Yes [c]
CPT	Yes [d]	Yes [e]	CPT/MAJ with ≥ 8 YOS [f]	Yes	No [g]	Yes	Yes	No	Yes [h]	Yes [f]
LT	No	Yes [i]	Yes [j]	Yes	Yes [k]	Yes	Yes [l]	No	Yes [m]	Yes [i] [j]

[a] *Other constraint types not mentioned in this table are:* flow conservation, control of input, and key arc relationship. The one flow conservation constraint (GOZO) applies to all grade segments. Of the two control of input constraints, one (TOTAUTH) applies to all grade segments, while the other (UBSG) applies to field grades only. The key arc relationship constraint (UR) applies to all segments except LT.

[b] Promotions to 0-7 are included as a type of attrition from the grade of 0-6, thus Y arcs (which represent promotees) are not required.

[c] Y arcs represent promotees to next higher grade. Promotions can occur at any node beginning at T_0 and ending at $T_{(NYRS-1)}$; at T_0, promotions result in reassignment to the alternate specialty, while in all other years, the reassignment to the alternate specialty is not effected until the completion of one tour length.

[d] The CPT segment is the only segment in which a W arc connects a specialty n node at T'_0 and a specialty n node at T_0 (e.g. arc W_{02121} connects node 21 at T'_0 to node 21 at T_0).

[e] X arcs represent captains with less than 8 YOS and are *only constructed for BES, where "from" and "to"* specialties are the same, up to year where earliest year group at T_0 would reach the beginning of the 8th YOS.

[f] Y arcs represent captains and majors with 8 or more YOS. Promotion from captain to major is considered in computing the attrition rates applied to the flows in the Y arcs; however, promotees remain in the Y arcs.

[g] The unfilled higher grade requirements (which normally make up the b-coefficient of the CREQ constraint) are used as lower bounds on the flows into the **BES** nodes. *The flows into the* **AES** nodes are influenced by lower bounds determined from attrition rates (normally higher than the unfilled higher grade requirements).

[h] Constraints are of the "less than or equal to" type as long as designations of alternate specialties can occur; thereafter, they are equalities.

[i] X arcs represent a combined second lieutenant (2LT) and first lieutenant (1LT) population. Promotion from 2LT to 1LT is considered in computing the attrition rates which apply to the X arcs.

[j] *Y arcs represent* LTs promoted to CPT since T_0.

[k] *The unfilled higher grade requirements* (which normally make up the b-coefficient of the CREQ constraint) are used as upper bounds on the flows in the Y_{mn} (m≠n) arcs to the point where the first year group of lieutenants at T_0 would attain 8 YOS. *Thereafter,* the CREQ constraints apply.

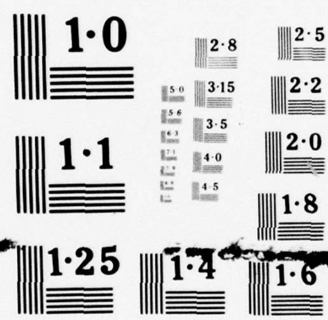
[l] CINC constraints are constructed from the point where the first year group of lieutenants at T_0 would attain 8 YOS, *up to* T_{NYRS}.

[m] Used only when a designee (LT/CPT with > 8 YOS) is due to return from alternate specialty, and then it is a "less than or equal to" *type*.

AD-A040 832 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 5/9
OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS) DOCUMENTATION.(U)
MAY 77 J D THOMAS, G E ARMSTRONG

UNCLASSIFIED CAA-D-77-1 NL

2 OF 5
ADA
040832



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

c. On-Line Inquiry Activity. - All printed outputs are optional in the on-line inquiry activity of the processing phase. By use of an appropriate set of MIRADS instructions, selected information on the status of a solution or solutions may be displayed on a computer terminal. If the terminal display is to be saved, then that image on the terminal display device can be printed. There are two general types of MIRADS statement sets: predefined and user-generated.

(1) The user may select a predefined set of MIRADS instructions, modify the predefined set, or compose a new set of instructions depending upon information needs. The predefined sets of instructions provide answers to the following types of questions:

(a) How many officers, by grade, should be in each preferred specialty pair?

(b) How many CPTs with 8 YOS should be designated specific alternate specialties?

(c) How many officers, by grade and specialty pairs, are assigned against a specialty requirement at a specified time?

(d) How much grade substitution is required for a given personnel policy?

Excerpts of reports answering these types of questions are at Figures V-14 through V-17, respectively.

(2) The user-generated type of MIRADS statement sets is limited only by MIRADS capabilities (explained in the MIRADS-2 Users Manual, published by Marshall Space Flight Center, NASA (Reference 3)).

d. Linkage Activity. - Two reports--the Unfilled Requirements Report and the ODSAS Procurement Report--are produced in the linkage activity of the processing phase. These two reports are described in subparagraphs (1) and (2), below, respectively.

(1) The Unfilled Requirements Report is produced upon completion of the processing of each segment. The report appears in two sequential parts (Figure V-18).

(a) The first part (upper portion of Figure V-18, titled REQUIREMENTS FOR GRADE N *** BEFORE *** DERIVING UNFILLED REQUIREMENTS), displays the force structure requirements for the current grade and the unfilled requirements (as determined in the previous ODSAS segment solution) for the next higher grade, as they were before the current grade segment was processed.

Primary specialty	Alternate specialty	Grade	Number of officers serving in alternate specialty[a/]
11	21	6	14
11	37	6	9
11	41	6	126
11	43	6	8
11	51	6	13
11	53	6	9
11	54	6	283
12	31	6	45
12	48	6	9
12	49	6	39
12	51	6	9
12	53	6	5
12	54	6	11
12	91	6	14
12	92	6	6
13	15	6	1
13	21	6	124
11	48	5	222
11	54	5	128
12	45	5	2
12	46	5	108
12	47	5	35
12	54	5	159
13	51	5	88
13	54	5	364
13	71	5	41
12	41	4	90
12	54	4	36
13	21	4	38

[a/]The number of officers serving in their primary specialty is also in this report. e.g., the first line of this report specifies that there are 14 colonels with specialty pair 11/21 serving in Specialty 21. Another line (not shown) would specify the number of colonels with specialty pair 11/21 serving in Specialty 11.

FIGURE V-14, Excerpt of Direct Access Information Retrieval Display for Number of Officers by Grade in Each Specialty Pair

Year	Primary specialty	Designated alternate specialty	Number of captains with 8 YOS that were designated alternate specialties
0	11	21	233
0	11	43	15
0	11	48	111
0	11	71	54
0	11	97	21
0	12	15	19
0	12	35	87
0	12	46	39
0	13	21	99
0	13	31	16
0	13	54	159
0	13	91	37
0	13	92	85
0	14	49	50
…	…	…	…
0	92	72	29
0	93	82	4
0	95	97	24
0	97	95	17
1	11	15	33
1	11	28	13
1	11	31	193
1	11	35	62
1	11	47	80
1	12	47	90
1	12	53	37
1	13	15	59
1	13	92	287
1	14	49	66
…	…	…	…
1	92	83	31
1	93	83	3
1	95	51	21
1	97	91	15
2	11	28	55
2	11	54	376

FIGURE V-15, Excerpt of Direct Access Information Retrieval Display of Specialty Designations for Captains

Grade	Time period	Alternate specialty	Primary specialty	Number assigned to requirement for primary specialty
4	1	11	11	1879 a/
4	1	15	11	67
4	1	21	11	164
4	1	28	11	2
4	1	37	11	4
4	1	42	11	170
4	1	43	11	7
4	1	46	11	10
4	1	48	11	80
4	1	49	11	13
4	1	51	11	24
4	1	53	11	79
4	1	54	11	180
4	1	71	11	4
4	1	92	11	17
4	1	93	11	9
4	1	97	11	22
5	1	11	11	520a/
5	1	28	11	4
5	1	41	11	44
5	1	42	11	9
5	1	46	11	18
5	1	47	11	23
5	1	48	11	52
5	1	49	11	22
5	1	53	11	16
5	1	54	11	13
5	1	54	11	28
6	1	11	11	256a/
6	1	21	11	2
6	1	37	11	1
6	1	41	11	43
6	1	43	11	1
6	1	48	11	15
6	1	51	11	1

a/ Indicates officers who remained in Specialty 11 from time period 0 to time period 1. Their alternate specialty is not shown in this report.

FIGURE V-16, Excerpt of Direct Access Information Retrieval Display for Number of Officers by Grade and Specialty Pairs Assigned to Specialty 11

Year	Grade	Specialty	Number of Officers grade-substituted in specialty requirements[a]
0	6	15	1
0	6	26	2
0	6	27	46
0	6	35	4
0	6	36	3
0	6	42	2
0	6	45	9
0	6	95	5
0	6	97	9
0	5	13	61
0	5	14	73
0	5	15	21
0	4	28	24
0	4	35	8
0	4	36	34
0	4	37	33
0	4	97	106
1	6	11	21
1	6	12	9
1	6	13	11
1	6	14	5
1	4	95	91
1	4	97	99
2	6	11	60
2	6	12	22
2	6	13	38
2	6	14	17

[a] Positions filled by grade substitution are filled by officers of the next lower grade with the same specialty as that required by the force structure.

FIGURE V-17, Excerpt of Direct Access Information Retrieval Display of Grade Substitution by Specialty Within Grade and Year

REQUIREMENTS FOR GRADE 6 ... BEFORE ... DERIVING UNFILLED REQUIREMENTS

	T0	T1	T2	T3	T4	T5	T6	T7	T8
SPECIALTY 11 REQUIREMENTS	283	281	281	281					
SPECIALTY 12 REQUIREMENTS	111	113	113	113					
SPECIALTY 13 REQUIREMENTS	175	172	172	167					
SPECIALTY 14 REQUIREMENTS	75	70	68	67					
SPECIALTY 15 REQUIREMENTS	4	4	4	4					
SPECIALTY 21 REQUIREMENTS	231	225	226	226					
SPECIALTY 25 REQUIREMENTS	45	41	41	41					
SPECIALTY 26 REQUIREMENTS	42	43	43	43					
SPECIALTY 27 REQUIREMENTS	42	42	43	43					
SPECIALTY 28 REQUIREMENTS	3	3	3	3					
SPECIALTY 31 REQUIREMENTS	45	45	45	45					
SPECIALTY 35 REQUIREMENTS	92	91	90	90					
SPECIALTY 36 REQUIREMENTS	33	29	29	29					
SPECIALTY 37 REQUIREMENTS	42	41	41	41					

REQUIREMENTS FOR GRADE 6 ... AFTER ... DERIVING UNFILLED REQUIREMENTS

	T0	T1	T2	T3	T4	T5	T6	T7	T8
SPECIALTY 11 REQUIREMENTS	0	67	94	136					
SPECIALTY 12 REQUIREMENTS	0	27	36	52					
SPECIALTY 13 REQUIREMENTS	0	41	59	80					
SPECIALTY 14 REQUIREMENTS	0	17	24	35					
SPECIALTY 15 REQUIREMENTS	0	1	2	2					
SPECIALTY 21 REQUIREMENTS	0	54	80	115					
SPECIALTY 25 REQUIREMENTS	1	10	13	18					
SPECIALTY 26 REQUIREMENTS	0	10	14	20					
SPECIALTY 27 REQUIREMENTS	0	10	15	21					
SPECIALTY 28 REQUIREMENTS	0	1	1	1					
SPECIALTY 31 REQUIREMENTS	0	11	15	22					
SPECIALTY 35 REQUIREMENTS	35	38	42	48					
SPECIALTY 36 REQUIREMENTS	1	7	10	14					
SPECIALTY 37 REQUIREMENTS	0	10	14	21					
SPECIALTY 88 REQUIREMENTS	0	1	2	2					
SPECIALTY 91 REQUIREMENTS	0	17	23	34					
SPECIALTY 92 REQUIREMENTS	0	27	30	46					
SPECIALTY 93 REQUIREMENTS	40	47	53	56					
SPECIALTY 95 REQUIREMENTS	0	19	27	39					
SPECIALTY 97 REQUIREMENTS	0	24	32	47					

FIGURE V-18, Excerpt of Unfilled Requirements Report

(b) The second part (lower portion of Figure V-18, titled REQUIREMENTS FOR GRADE N *** AFTER *** DERIVING UNFILLED REQUIREMENTS), displays the combined total of the unfilled requirements remaining for the current and higher grade, after the current segment solution values for the number of filled requirements have been considered in deriving the unfilled requirements.

(c) Derivation of unfilled requirements by ODSAS involves more than simply finding the difference between the capacity and the activity at a node; attrition of officers assigned at T_i has to be taken into account when computing the unfilled requirements at T_{i+1}.

(2) The ODSAS Procurement Report is produced only once--upon completion of the LT segment.

(a) The requirements in this report are the unfilled requirements at T_1 derived from the LT segment solution. Since actual procurement of new officers is by BES only, the AES requirements must be prorated to the several BES. This allocation is accomplished according to the relationships of BES/AES specialty pairs to AES requirements. These relationships were derived in the CPT and MAJ segment solutions.

(b) The report appears in two parts, with a third (optional) part available if desired by the user. The report format (see Figure V-19) is the same for all three parts. The two leftmost columns pertain to BES only; all BES are identified in the column headed SPECIALTY NUMBER, and the requirements for each BES appear in the column headed BES REQMT. The next 16 columns, headed by the AES numbers, contain the AES requirements prorated to the respective BES. Requirements for each BES and the AES requirements prorated to that BES are summed across each line and the result appears in the rightmost column (headed TOTAL REQMT). Each data column is also summed and the resulting total appears on the bottom line (labeled TOTAL). The number at the extreme lower right of the report is the grand total of all BES and AES requirements, and is the same whether summed horizontally (the TOTAL line) or vertically (the TOTAL REQMT column).

1. Part 1 of 2 (Figure V-19) displays, for each BES, the total number of officers to procure, considering the requirements for the BES plus the prorated number of officers needed to meet the AES requirements.

ODSAS PROCUREMENT REPORT PART 1 OF 2
INCLUDES PRORATION OF AES TO ALL BASIC ENTRY SPECIALTIES

SPECIALTY NUMBER	BES REQMT	****** 15	41	45	46	47	48	49	51	52	53	54	86	91	93	95	**** 97	TOTAL REQMT
		PRORATED AES REQUIREMENTS																
11	1181.	0.	3.	3.	8.	3.	20.	1.	13.	0.	1.	7.	0.	0.	1.	0.	0.	1240.
12	618.	7.	2.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	629.
13	759.	7.	102.	1.	0.	0.	1.	1.	1.	0.	2.	5.	0.	0.	0.	0.	0.	878.
14	801.	79.	4.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	8[illegible]6.
21	577.	0.	2.	0.	0.	0.	2.	2.	0.	2.	22.	0.	0.	39.	0.	0.	0.	646.
25	369.	0.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	371.
26	82.	0.	2.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	85.
27	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	6.	14.
28	14.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	18.
31	333.	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	341.
35	81.	21.	5.	1.	0.	0.	0.	0.	1.	0.	1.	5.	0.	0.	0.	0.	0.	115.
36	151.	0.	10.	0.	0.	16.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	178.
37	54.	0.	0.	1.	0.	0.	0.	0.	5.	0.	0.	4.	0.	0.	0.	0.	0.	65.
42	220.	16.	0.	11.	0.	2.	0.	0.	0.	0.	1.	0.	4.	0.	0.	1.	0.	255.
43	9.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	9.
44	124.	0.	0.	0.	0.	0.	0.	0.	0.	0.	17.	0.	0.	0.	0.	0.	0.	142.
71	73.	0.	2.	0.	0.	0.	0.	7.	0.	0.	1.	0.	0.	0.	0.	0.	0.	83.
72	38.	0.	0.	0.	0.	0.	0.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	42.
73	31.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	0.	0.	0.	0.	0.	34.
74	73.	0.	1.	1.	0.	0.	0.	0.	1.	0.	0.	2.	0.	0.	0.	0.	0.	78.
75	75.	0.	17.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	94.
76	20.	0.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	1.	0.	0.	0.	24.
77	169.	0.	0.	0.	0.	0.	0.	0.	5.	0.	0.	0.	0.	0.	0.	0.	0.	175.
81	26.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	26.
82	13.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	13.
83	5.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	6.
87	32.	0.	2.	0.	0.	0.	0.	0.	0.	0.	9.	0.	2.	0.	0.	0.	0.	44.
88	257.	0.	2.	0.	0.	0.	0.	0.	0.	0.	0.	1.	1.	0.	0.	0.	0.	260.
92	170.	511.	6.	1.	0.	0.	0.	0.	1.	0.	1.	0.	9.	2.	24.	2.	61.	789.
TOTAL	6362.	648.	162.	18.	13.	23.	24.	15.	29.	2.	64.	25.	15.	42.	27.	4.	67.	7540.

FIGURE V-19, Example of ODSAS Procurement Report, Part 1 (BES Requirements, Plus AES Requirements Prorated to All Specialties)

2. Part 2 of 2 (Figure V-2Ø) is similar to Part 1, except that the number of officers needed to meet the AES requirements is prorated to the non-combat arm specialties only (i.e., the total requirements for the combat arms specialties are equal to the BES REQMT).

3. Option 1 (Figure V-21) displays the results of a modified version of Part 2, wherein one or more AES (specified by the user) are prorated to both the combat and non-combat arm specialties, while all remaining AES are prorated among the non-combat arm specialties only.

ODSAS PROCUREMENT REPORT PART 2 OF 2
EXCLUDES PRORATION OF AES TO ALL COMBAT ARMS SPECIALTIES
PRORATED AES REQUIREMENTS

SPECIALTY NUMBER	BES REQMT	****** 15	41	45	46	47	48	49	51	52	53	54	86	91	93	95	**** 97	TOTAL REQMT
11	1181.																	1181.
12	618.																	618.
13	759.																	759.
14	801.																	801.
21	577.	0.	5.	0.	0.	0.	13.	3.	0.	2.	24.	0.	0.	39.	0.	0.	0.	664.
25	369.	0.	6.	0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	377.
26	82.	0.	5.	0.	0.	0.	2.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	91.
27	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	6.	14.
28	14.	0.	0.	0.	12.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	26.
31	333.	8.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	342.
35	81.	24.	17.	1.	0.	0.	3.	0.	1.	0.	1.	10.	0.	0.	0.	0.	0.	138.
36	151.	0.	33.	0.	0.	19.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	205.
37	54.	0.	0.	1.	0.	1.	0.	0.	10.	0.	0.	8.	0.	0.	0.	0.	0.	73.
42	220.	19.	0.	13.	1.	3.	0.	0.	0.	0.	1.	0.	4.	0.	0.	1.	0.	261.
43	9.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	9.
44	124.	0.	0.	0.	0.	0.	0.	0.	0.	0.	19.	0.	0.	0.	0.	0.	0.	143.
71	73.	0.	5.	0.	0.	0.	0.	8.	0.	0.	1.	0.	0.	0.	0.	0.	0.	87.
72	38.	0.	0.	0.	0.	0.	0.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	42.
73	31.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	0.	0.	0.	0.	0.	34.
74	73.	0.	3.	1.	0.	0.	2.	0.	1.	0.	0.	5.	0.	0.	0.	0.	0.	85.
75	75.	0.	55.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	132.
76	20.	0.	0.	0.	0.	0.	0.	0.	5.	0.	0.	0.	0.	1.	0.	0.	0.	26.
77	169.	0.	0.	0.	0.	0.	2.	0.	10.	0.	0.	0.	0.	0.	0.	0.	0.	181.
81	26.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	26.
82	13.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	13.
83	5.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	6.
87	32.	0.	5.	0.	0.	0.	0.	0.	0.	0.	10.	0.	2.	0.	0.	0.	0.	49.
88	257.	0.	5.	0.	0.	0.	0.	0.	0.	0.	0.	2.	1.	0.	0.	0.	0.	265.
92	170.	597.	21.	1.	0.	0.	0.	0.	1.	0.	1.	0.	9.	2.	25.	2.	61.	891.
TOTAL	6362.	648.	162.	18.	13.	23.	24.	15.	29.	2.	64.	25.	15.	42.	27.	4.	67.	7540.

FIGURE V-20, Example of ODSAS Procurement Report, Part 2 (BES Requirements, Plus AES Requirements Prorated to Non-Combat Arms Specialties Only)

ODSAS PROCUREMENT REPORT
***OPTION 1 ***

SPECIALTY NUMBER	BES REQMT	****** 15	41	45	46	47	48	PRORATED AES REQUIREMENTS 49	51	52	53	54	86	91	93	95	**** 97	TOTAL REQMT
11	1181.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1181.
12	618.	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	625.
13	759.	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	766.
14	801.	79.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	880.
21	577.	0.	5.	0.	0.	0.	13.	3.	0.	2.	24.	0.	0.	39.	0.	0.	0.	664.
25	369.	0.	6.	0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	377.
26	82.	0.	5.	0.	0.	0.	2.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	91.
27	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	6.	14.
28	14.	0.	0.	0.	12.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	26.
31	333.	7.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	341.
35	81.	21.	17.	1.	0.	0.	3.	0.	1.	0.	1.	10.	0.	0.	0.	0.	0.	135.
36	151.	0.	33.	0.	0.	19.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	205.
37	54.	0.	0.	1.	0.	1.	0.	0.	10.	0.	0.	8.	0.	0.	0.	0.	0.	73.
42	220.	16.	0.	13.	1.	3.	0.	0.	0.	0.	1.	0.	4.	0.	0.	1.	0.	258.
43	9.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	9.
44	124.	0.	0.	0.	0.	0.	0.	0.	0.	0.	19.	0.	0.	0.	0.	0.	0.	143.
71	73.	0.	5.	0.	0.	0.	0.	8.	0.	0.	1.	0.	0.	0.	0.	0.	0.	87.
72	38.	0.	0.	0.	0.	0.	0.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	42.
73	31.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	0.	0.	0.	0.	0.	34.
74	73.	0.	3.	1.	0.	0.	2.	0.	1.	0.	0.	5.	0.	0.	0.	0.	0.	85.
75	75.	0.	55.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	132.
76	20.	0.	0.	0.	0.	0.	0.	0.	5.	0.	0.	0.	0.	1.	0.	0.	0.	26.
77	169.	0.	0.	0.	0.	0.	2.	0.	10.	0.	0.	0.	0.	0.	0.	0.	0.	181.
81	26.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	26.
82	13.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	13.
83	5.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	6.
87	32.	0.	5.	0.	0.	0.	0.	0.	0.	0.	10.	0.	2.	0.	0.	0.	0.	49.
88	257.	0.	5.	0.	0.	0.	0.	0.	0.	0.	0.	2.	1.	0.	0.	0.	0.	265.
92	170.	511.	21.	1.	0.	0.	0.	0.	1.	0.	1.	0.	9.	2.	25.	2.	61.	805.
TOTAL	6362.	648.	162.	18.	13.	23.	24.	15.	29.	2.	64.	25.	15.	42.	27.	4.	67.	7540.

FIGURE V-21, Example of ODSAS Procurement Report, Option 1 (Part 2, Modified by Proration of Requirements for Specified AES to Both Combat and Non-Combat Arms Specialties)

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER VI
ODSAS OPERATIONS GUIDE

1. Purpose. - The purpose of this chapter is to provide the operating instructions for ODSAS. The system is designed to be run on the UNIVAC 11Ø8 at MILPERCEN. These operating instructions can best be utilized on a demand-type terminal (e.g., UNISCOPE 1ØØ); however, they could also be punched into cards and submitted as a batch run.

2. Catalogued Runstreams. - The instructions for the user to operate the system are composed of a minimum number of EXEC 8 control language statements (also called EXEC 8 commands) because of the use of catalogued runstreams. Through this technique, one EXEC 8 command triggers a sequence of EXEC 8 statements that were previously placed, or catalogued, in an element within a program file. All the catalogued runstreams are located on a disc file named "PFCAA."; the contents of each runstream is displayed in Chapter IX. Therefore, the user need only know a few simple EXEC 8 commands in order to execute a catalogued runstream. They are:

a. @START PFCAA.element name. - This is for a longer running program(s) and is treated as a batch job by the computer operating system.

b. @ADD,L PFCAA.element name. - This is for a program that takes relatively little computer time to execute and is treated as a demand job by the computer operating system.

3. Operating Instructions for Initialization Phase

a. Procedure for SACS Tape Translation. - Since the SACS tape is produced at USAMSSA on an IBM computer, that tape has to be translated so that it can be read on System 2 at MILPERCEN (System 2 was designated as the computer to be used for ODSAS). The translation is done on System 3 because that is the only system that has the necessary translation capability (i.e., EBCDIC-FIELDATA).

(1) The SACS tape is classified CONFIDENTIAL; therefore, the translation run ("SACSEXTRACT") has to be scheduled with the MILPERCEN Customer Services Division (CSD). The user should contact CSD when the "SACSEXTRACT" run (a listing and description is

in Chapter IX) is required. CSD will prepare a Production Control Sheet (DAPC-PS-Form 2), and provide a copy to the user.

(2) The user must prepare two data cards for the SACSEXTRACT card deck and insert them immediately following the "@XQT PFODSAP.SACSEXTRACT" card. The two data cards are as follows:

(a) Card 1 - the number of tape reels (e.g., 2), punched in card Column 1.

(b) Card 2 - the reel numbers (e.g., HFØ4Ø, HFØ41). The first reel number is punched, left-justified, in card Column 1. The second reel number, if any (the program will accept up to 6 reel numbers), is left-justified in card Column 7 (no comma between the reel numbers).*

(3) The user then submits the SACSEXTRACT card deck (as a batch run) and the Production Control Sheet, along with other forms listed below, to the Computer Operations Division before 133Ø on the desired day/evening of the run:

(a) Job Series Sheet (DAPC-PS-Form 4ØØ)

(b) ADP Job Instructions Sheet (DAPC-PS-Form 4Ø1-1)

(c) Data Control Sheet (DAPC-PS-Form 58)--to be used in the event the specified input is not available.

(4) A copy of the runstream is kept on the PFCAA file in an element named "SACSEXTRACT" on System 2, in case the card deck is lost, damaged, or destroyed.

(5) The translated tape is unclassified since the unit identification code (UIC) is not associated with the authorized strength.

b. Completing the Initialization Phase. - Once the tape is translated and the information extracted, then four catalogued runstreams have to be executed to finish the initialization phase of ODSAS. The necessary EXEC 8 control language statements to accomplish the initialization phase activities are:

---

*The user is responsible for assuring that the correct SACS tape(s) is being used. Reel numbers for the most recent PERSACS DETAIL FILE RECORD can be obtained from the Military Strength Systems Branch, HQ DA Military Systems Division, PERSINS-D.

(1) <u>@START PFCAA.SACS</u>

(a) The PFCAA.SACS runstream must be modified with the ED processor to reflect the starting date of the network to be analyzed. This date, entered in YYMMDD format, is normally the first day of a fiscal year (e.g., 771001)* and immediately follows the "@XQT PFODSAP.SACSCREATE" statement in the runstream as illustrated below.

```
   @RUN,/TP OD001S,090112,CAA,300,500
   @TYPE 2,BA
   @DELETE,C OD001UD90
   @PRT,I
   @ASG,UP OD001UD90
   @BRKPT PRINT$/OD001UD90
   @HDG,P SACS FILE CREATION FOR ODSAS     UNCLASSIFIED
   @ASG,T TAPE.,16N,08835              . I/D TAPE FILE
   @DELETE,C TEMPDISC.
   @ASG,UP TEMPDISC.,F/9/PO5/50        . TEMP DISC FILE
   @USE 10.,TEMPDISC.
   @COPY,G TAPE.,10
   @REWIND TAPE.
   @FREE TAPE.
   @DELETE,C ODOUTUD01.
   @PRT,I
   @ASG,UP ODOUTUD01.,F//POS/300
   @USE 11.,ODOUTUD01
   @XQT PFODSAP.SACSPREPRO
   @PMD,EL
   @FREE,A 10.
   @FREE,A 11.
   @PRT,F ODOUTUD01.
   @USE 10.,ODOUTUD01.
   @ASG,A ODSAPUD10.
   @USE 11.,ODSAPUD10.
   @PRT,I
   @XQT  PFODSAP.SACSCREATE
→  771001
```

---

*The first day of the coming FY (rather than the last day of the current FY) is used because of discontinuity in force accounting in the SACS data (e.g., total force requirements on the last day of FY 77 and the first day of FY 78 may be significantly different because of the phaseout of units/positions as of the end of the FY).

```
@PMD,E
@PRT,F ODSAPUD1Ø.
@PRT,I
@BRKPT PRINT$
@FREE
@SYM,SU ODØØ1UD9Ø.,,PR
@ASG,A ODØØ1UD9Ø.
@ASG,A PFPRINT.
@COPY,I ODØØ1UD9Ø.,PFPRINT.PRINT
@END
```

(b) The EXEC 8 statements to make the change are as follows:

```
@ED,U PFCAA.SACSCREATE

LOCATE @XQT PFODSAP.SACSCREATE

NEXT

CHANGE /771ØØ1/        (Type in desired date in
                        YYMMDD format)

EXIT
```

(c) This runstream creates a positional requirements file, ODSAPUD1Ø, and the positional requirements report. After user verification of the positional requirements report, ODSAPUD1Ø must be copied to the master positional requirements file, ODSACUDØ2, via the following sequence of commands:

```
@CHG,Z ODSACUDØ2.            REMOVE READ-ONLY MODE

@COPY ODSAPUD1Ø., ODSACUDØ2.

@CHG,V ODSACUDØ2.            RE-SET READ-ONLY MODE
```

(d) Prior to the processing of this run, the tape library must be notified of the need for the translated tape (produced in the SACSEXTRACT run) to be available on System 2.

(2) @START PFCAA.TOUR. - The card-image input file, ODTURUDØ1 (described in Chapter IV, paragraph 2a) must be prepared before this runstream is processed. This runstream creates, or updates, the specialty preferences file, ODRATUDØ1.

(3) @START PFCAA.INITIAL. - The card-image input file, ODR8SUDØ1 (described in Chapter IV, paragraph 2b) must be prepared before this runstream is processed. This runstream creates the

attrition and promotion rates file, ODPOPUDØ1, the master input parameter file, ODINPUDØ2, needed to begin the first system segment, and the attrition and promotion rates report.

(4) @ADD,L PFCAA.START. - This runstream copies the master specialty requirements file, ODSACUDØ2, and the master input parameter file, ODINPUDØ2, to the corresponding files, ODSACUDØ1 and ODINPUDØ1, respectively, in the processing phases. This runstream also initializes the data base files.

(5) The runstreams in (1), (2), and (3) above can be initiated from the demand terminal in any sequence, and will be processed as batch jobs. The fourth runstream (in (4) above) takes very little computer time, so it is executed in demand mode.

4. Operating Instructions for Processing Phase. - The EXEC 8 control language statements to operate all of the processing phases are also simple statements which refer to catalogued runstreams. The instructions depend upon whether any of the field grade segments are to be segmented within grade, and the user's desire to utilize the on-line inquiry capability offered by MIRADS.

a. Standard Instructions for Segmented or Unsegmented Processing. - At Table VI-1 are all the EXEC 8 standard control language statements needed to run the processing phases of ODSAS. If segmentation within grade was selected for any of the field grades, then the statements in the column labeled "Segmentation Within Grade" should be used for the grades selected for segmentation. If segmentation within grade was not selected for a field grade, then the statements in the column labeled "Unsegmented" should be used for those grades. The EXEC 8 control statements that begin "@START" are entered through the demand terminal and are processed as batch jobs, thus the output will probably not be immediately available. The processing time for each runstream depends upon the number of rows in the LP problem to be solved. The standard set includes on-line inquiry at the completion of each segment. The on-line inquiry capability can be selectively bypassed by deleting the statements as indicated by the footnote. A possible future enhancement would be to produce standard reports via MIRADS within a new catalogued runstream. That runstream name would then either replace or supplement the statement indicated by the footnote.

b. Optional Instructions for Rerunning. - An optional runstream, PFCAA.SAVECOPY/TOTAPE, may be @ADD'd after any @START PFCAA. RUN-__ __, and before @ADD,L PFCAA.LINK-MODULE. This runstream will copy the current version of eleven ODSAS disc files to tape for retention and use in the event that the next

TABLE VI-1, EXEC 8 Commands for Processing Phase of ODSAS
(continued on next page)

Grade	Mode of processing: Segmentation within grade	Mode of processing: Unsegmented
6	@START PFCAA.RUN-COL1 (submitted as batch job from demand terminal)	@START PFCAA.RUN-COLXX (submitted as batch job from demand terminal)
	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)
	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)	
	@START PFCAA.RUN-COL2 (submitted as batch job from demand terminal)	
	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	
5	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)
	@START PFCAA.RUN-LTC1 (submitted as batch job from demand terminal)	@START PFCAA.RUN-LTCY (submitted as batch job from demand terminal)
	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)
	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)	
	@START PFCAA.RUN-LTC2 (submitted as batch job from demand terminal)	

TABLE VI-1, EXEC 8 Commands for Processing Phase of ODSAS
(continued on next page)

Grade	Mode of processing: Segmentation within grade	Mode of processing: Unsegmented
5 (cont)	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	
4	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)
	@ADD,L PFCAA.SEP-MODULE/4 (separates COL from LTC records in cumulative data base)	@ADD,L PFCAA.SEP-MODULE/4 (separates COL from LTC records in cumulative data base)
	@START PFCAA.RUN-MAJ1 (submitted as batch job from demand terminal	@START PFCAA.RUN-MAJZZ (submitted as batch job from demand terminal)
	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)
	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)	
	@START PFCAA.RUN-MAJ2 (submitted as batch job from demand terminal)	
	@ADD,L PFCAA.MIRADS[a/] (user interrogates data bases in demand mode)	
3	n/a	@ADD,L PFCAA.LINK-MODULE (updates files for next segment)

TABLE VI-1, EXEC 8 Commands for Processing Phase of ODSAS
(continued on next page)

Grade	Mode of processing	
	Segmentation within grade	Unsegmented
3 (cont)	n/a	@ADD,L PFCAA.SEP-MODULE/3 (separates LTC from MAJ records in cumulative data base) START PFCAA.RUN-CPT (submitted as batch job from demand terminal) @ADD,L PFCAA.MIRADS[a] (user interrogates data bases in demand mode)
2	n/a	@ADD,L PFCAA.LINK-MODULE (updates files for next segment) @ADD,L PFCAA.SEP/MODULE/2 (separates MAJ from CPT records in cumulative data base) @START PFCAA.RUN-LT (submitted as batch job from demand terminal) @ADD,L PFCAA.MIRADS[a] (user interrogates data bases in demand mode) @ADD,L PFCAA.LINK-MODULE (produces final reports for ODSAS) @ADD,L PFCAA.SEP-MODULE/1 (puts LT records on cumulative data base I/P file with all other grade records and

TABLE VI-1, EXEC 8 Commands for Processing Phase of ODSAS (concluded)

Grade	Mode of processing	
	Segmentation within grade	Unsegmented
2 (cont)	n/a	implements cumulative data base containing all grade records)
		@ADD,L PFCAA.MIRADS (user interrogates data bases in demand mode - the last query set exercised must be the PROCUREMENT standard query set (see Table VI-2))
		@ADD,L PFCAA.PROCURE (produces standard and optional ODSAS Procurement Reports)

[a] Statement may be omitted if on-line inquiry is not desired.

processing phase in sequence should need to be re-run. If a processing phase needs to be re-run after subsequent processing phases have been run (in other words, if any of the eleven files have been updated), then the runstream PFCAA.SAVECOPY/TODISC should be @ADD'd, followed by @ADD,L PFCAA.LINK-MODULE. These latter two runstreams will restore the disc files to the point where a processing phase can be restarted (i.e., type in a runstream of the form @START PFCAA RUN-_ _ _ _).*

5. Procedures to Ascertain the State of Processing of the System Segments. - During the processing of a system segment, two messages on the processing status are placed in the MAIL file which

---

*Prior to saving or retrieving the 11 ODSAS files (via the SAVECOPY/TOTAPE or SAVECOPY/TODISC runstreams respectively) an ADP Job Instruction Sheet, DAPC-PS Form 4Ø1, dated 16 May 73, must be submitted to PERSINS-D ASP Branch (DAPC-PSO-D) to ensure that the appropriate tape (a blank tape is required for SAVECOPY/TOTAPE, specific reel number of a previously saved data base file is required for SAVECOPY/TODISC) is mounted on System 2.

can be accessed with the ED processor. The messages are sent when the FMPS processing is begun and when the processing of a segment is concluded. Since the FMPS solution activity is the longest running and uses the most computer resources, it is at the beginning and end of this critical processing activity that the messages are sent. As mentioned above, the MAIL file can be accessed with the ED processor by editing any ODSAS data or program file. This can be accomplished by using the demand terminal to enter a command similar to the following:

@ED,R ODINPUDØ1

(R is "read-only" mode, ODINPUDØ1 could be any ODSAS file name.)

The following message will then be displayed on the CRT:

DO YOU WANT YOUR MAIL?

The user should respond by typing in:

YES

-or-

NO

and transmitting the response. If the response is "YES," the following two types of messages will appear on the CRT:

FMPS MODULE ENTERED FROM: ODØØ6A TIME: hhmmss	first type
FMPS-MODULE COMPLETED FROM: ODØØ6A TIME: hhmmss	second type

The last two characters in the run-ID following "From:" (shown as "6A" in the example) is the unique identifier for the segment from which the message was sent; e.g., ODØØ6A is the run-ID of COL segment 1 (see Chapter IX for unique run-IDs). The "hhmmss" value is the time the message was sent. The MAIL file contains as many as 10 messages; however, once the MAIL is displayed or a "NO" response is transmitted, the file is emptied. If the "DO YOU WANT YOUR MAIL" system response does not appear, then processing has most likely not begun for any segment since the last time the MAIL file

was read (the exception would be if the commands preceding the FMPS module were being processed at the time of inquiring for the MAIL, thus processing had begun, but not to the point of generating a message).

6. Procedures to Recover from Premature Termination of System Segment. - If any system segment terminates prematurely during the FMPS solution activity because of reasons other than an infeasible solution, then one of the following two runstreams can be executed (after a modification, if needed, is made to the control statements within the runstream):

@START PFCAA.RECOVER for COL, LTC or MAJ segments

or

@START PFCAA.RECOVER/CPT-LT for CPT or LT segments

The first portion of the two runstreams is identical (see reproduction below), and includes a coded two-digit entry (indicated by the arrow) that identifies the segment being processed.

```
@RUN ODØØ1, Ø9Ø112,CAA
@ASG,A ODØØ1UD9Ø.
@BRKPT PRINT$/ODØØ1UD9Ø
@HDG,P ODSAS RECOVERY OF FMPS RUN
@ASG,A PFFMPS.
@ED,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED, I SEGMENT
→ 61
@ASG,A ODDBSUD1.
.
.
.
```

The one modification that may have to be made before attempting to recover the segment processing is to change the two-digit segment identifier. The segment identifier code numbers are as follows:

6Ø - COL unsegmented
61 - COL segment 1
62 - COL segment 2
5Ø - LTC unsegmented
51 - LTC segment 1
52 - LTC segment 2

40 - MAJ unsegmented
41 - MAJ segment 1
42 - MAJ segment 2
30 - CPT segment
20 - LT segment

This change is accomplished by typing in the following EXEC 8 statements at the demand terminal:

@ED,U PFCAA.RECOVER

12 (meaning go to line 12)

Change/61/nn/ (meaning change segment identification from 61 to nn (nn = 60, 61, 62...20, as shown above)

This runstream will begin processing the LP problem at the point where the processing results were last saved, and perform all other functions of the original runstream.

7. <u>Procedures for Modifying the LP Problem for Re-running</u>. - If the segment just completed has to be re-run because, through analysis of the solution, some input to the system was erroneous, or missing, then one or more of the following procedures can be performed depending upon the error or omission.

a. <u>To Change an Attrition or Promotion Rate</u>. - If after analyzing the ODSAS solution, an alternate solution is desired by changing the weighted average attrition or promotion rates, then a catalogued runstream can be executed with the following EXEC 8 command to update specific rate(s):

@ADD,L PFCAA.UPDATE

This may be done in demand mode since it takes less than 1 minute to process. This runstream could also be used to correct any of the rates originally produced in the initialization phase by the runstream PFCAA.INITIAL, rather than re-run PFCAA.INITIAL. The changes are placed in "PFODSAP.UPDATE/CARDS" via the ED processor. The card formats for the update cards is shown in Chapter IV of this documentation manual.

b. <u>To Change the Specialty Preferences</u>. - The catalogued runstream, PFCAA.TOUR, is used to create, add, or change specialty preferences. In the initialization phase, it was used to create the ODPOPUD01. file; it can also be used to add to, or change, specialty preferences by inserting a parameter card calling for

the update mode of processing, followed by cards specifying the new or changed specialty preferences (see Chapter IV for card formats).

c. To Change Requirements Data for a Specialty. - The same runstream as in subparagraph a above can be used to change a specialty's requirements by grade and year. The PFCAA.UPDATE runstream should be used to change values computed in the PFCAA.SACS. runstream of the initialization phase if the FMPS activity has not been attempted. Otherwise, another catalogued runstream, called PFCAA.MODIFY, should be used. This employs an FMPS procedure that will accept the changed requirements values, and will re-solve the modified LP problem from an advanced basis, rather than from the initial input. The input format for the changed requirements values is listed in Chapter VI of the FMPS Programer's Reference Manual and Chapter IV of this manual.

8. Procedures for Saving the MIRADS Data Bases. - ODSAS creates two data bases for on-line inquiry--a data base of the last segment processed, and a cumulative data base which contains the results of all system segments. These data base files have unique names (CURSEG and CUMSEG, respectively) generated by the MIRADS programs and ordinarily would need to be saved and accounted for by the computer operations section. However, the MIRADS system provides an alternative procedure to save the data bases on tape, and recall them to a disc file when the user actually needs them for on-line inquiry.* To save the data base files, the following command (which references a catalogued runstream) should be used:

@ADD, L PFCAA.ROLLOUT

To restore the data base files to disc (from tape), the following command (also referencing a catalogued runstream) should be used:

@ADD, L PFCAA.ROLLIN

---

*Prior to saving or retrieving the data base files (via the ROLLOUT or ROLLIN procedures, respectively) an ADP Job Instruction Sheet, DAPC-PS Form 4Ø1, dtd 16 May 73, must be submitted to PERSINS-D ADP Branch (DAPC-PSO-D) to ensure that the appropriate tape (a blank tape is required for ROLLOUT, specific reel number of a previously saved data base file is required for ROLLIN) is mounted on System 2.

9. General Procedures for On-Line Inquiry of the Data Bases. - At the completion of each processing phase the CURSEG and CUMSEG data bases are available for on-line inquiry by the user. When the EXEC 8 command @ADD, L PFCAA.MIRADS is entered, a message is displayed on the CRT, asking for a file name of the data base to be referenced. The user should respond by typing in either "CURSEG" or "CUMSEG," depending upon the user's needs, and then transmitting the response. Assuming correct spelling, the system will display a message asking for a password. The user should then respond by typing in the word "ODSAS," and then transmitting. At this point the system will display the word READY, and on-line inquiry can begin. There are 12 predefined query sets (described in detail in Paragraph 10 and Table VI-2 below) that can be processed by the following type command:

DO "query set name"

Query sets can also be composed at the terminal and then processed by typing the word RUN. The standard query set names are defined and listed in Paragraph 10 below. When either of these MIRADS commands are transmitted (RUN or DO), the MIRADS system begins processing the query set and responds:

QUERY NOW PROCESSING

followed by

FILE CONTAINS nnnn RECORDS

QUERY SELECTED nn RECORDS

ENTER OUTPUT REPORT SITE ID

The user must respond to this last statement by entering one of the following commands via keyboard entry:

NONE - no output desired

PRINT = n (display first n records on CRT)
DRUM - write output to disc file

or

depress transmit key and all output will be displayed on CRT (and the on-site printer if it is selected)

10. Standard Query Sets for Data Base Access. - There are 12 standard query sets of MIRADS statements that are loaded when the data base(s) are loaded (in the PFCAA.IMPLEMENT catalogued run-stream) and are available for the user to employ as needed. These standard query sets are as shown in Table VI-2.

11. Modifications to Standard Query Sets. - As noted in some of the standard query sets listed in Table VI-2, the statements may be easily modified so that they can be used for any specialty or year, not just the ones specified in the examples. The MIRADS system has an EDIT function that enables the user to modify query sets, without exiting to the UNIVAC 11Ø8 ED processor and re-loading the revised query sets and data bases. The EDIT function with associated commands is very similar to the ED processor and can perform a selected subset of the ED processor functions. All of the capabilities of the EDIT function are explained in the MIRADS Users Manual; however, one of the capabilities, the CHANGE command, is the primary one used to modify the standard query sets. For example, using the NODEACTIVITY query set, the following sequence of commands may be used to transform the original query set concerning Specialty 11, to a modified query set concerning Specialty 49:

```
Original Query Set:  Q,(CYEAR = Ø AND CTO = 11 AND CACTIVITY
                       GT Ø AND CFROM GT Ø AND (CPREFIX = X
                       OR Y)) OR (CYEAR = 1 AND CFROM = 11 AND
                       CACTIVITY GT Ø)
                     S,CYEAR
                     C,CYEAR,$TOTAL = SUM ACTIVITY
                     P,CPREFIX,CYEAR, CFROM,CTO, CACTIVITY,$TOTAL
                       SP 1
```

Type in the following commands

```
EDIT NODEACTIVITY

CHANGE/CTO = 11/CTO = 49/ALL

CHANGE      /CFROM = 11/CFROM = 49/ALL

EXIT
```

The CHANGE command directs that every place where "CTO = 11" and "CFROM = 11" appear in the query set, "CTO = 49" and "CFROM = 49," respectively, should replace them. The resultant query set is shown below:

```
Changed Query Set:  Q,(CYEAR = Ø AND CTO = 49 AND CACTIVITY GT
                      Ø AND (CPREFIX = X OR Y)) OR (CYEAR = 1
                      AND CFROM = 49 AND CACTIVITY GT Ø
                    S,CYEAR
                    C,CYEAR,$TOTAL = SUM CACTIVITY
                    P,CPREFIX,CYEAR,CFROM,CTO CACTIVITY,$TOTAL
                      SP 1
```

TABLE VI-2, Standard Query Sets for Data Base Access (continued on next page)

Query set name	Applicable data base	Function	Sample statement listing
NODECAP	CURSEG, CUMSEG	Retrieves all row records for nodes which have input flows equal to one or both capacities, and displays the node identification (by year and specialty number), the amount of input and the capacity value, in specialty within year order.	Q,ID = R AND PREFIX = N AND YEAR GE Ø AND FROM GT Ø AND GRADE GT Ø AND ACTIVITY EQ *UPLIMIT C,ALL,$GRD = GRADE / 1Ø C,ALL,$IN-SPECIALTY = FROM S,YEAR,GRADE,D,FROM P,YEAR,$GRD,$IN-SPECIALTY,ACTIVITY,UPLIMIT
NOTCAPT-Ø	CURSEG, CUMSEG	Retrieves the row records for which the flow into a node at $T_{\emptyset}$ is less than total requirements capacity, and displays for each node, the specialty number, amount of flow into the node, the capacity and the percent of the capacity used by the input flow.	Q,ID = R AND PREFIX = N AND YEAR EQ Ø AND GRADE GT Ø AND FROM GT Ø AND ACTIVITY NE *UPLIMIT C,ALL,$IN-SPECIALTY = FROM C,ALL,$GRD = GRADE / 1Ø C,ALL,3,$FILL% = ACTIVITY / UPLIMIT S,YEAR,GRADE,D,FROM P,YEAR,$GRD,$IN-SPECIALTY,ACTIVITY,UPLIMIT,$FILL%
CPTDESIGN8	CURSEG	Retrieves all row records in the CPT or LT data bases which indicate alternate specialty designations for CPTs, and displays by specialties within year.	Q,ID = R AND PREFIX = V AND YEAR GE Ø AND ACTIVITY GT Ø S,YEAR,FROM,TO P,YEAR,FROM,ACTIVITY
SPEC-PAIRS	CURSEG	Retrieves all column records for W arcs in the $T_{\emptyset}$ - $T_{\emptyset}$ interval that have a value greater than zero, representing dual-qualified officers at $T_{\emptyset}$. The utilization ratio of the specialty pair, as a decimal, is also displayed.	Q,CID = C AND CPREFIX = W AND CACTIVITY GT Ø C,ALL,3,$RATIO = NRATIO / (RATIO + MRATIO) S,CFROM,CTO P,CFROM,CTO,CACTIVITY,$RATIO
CUM-SP-PAIRS	CUMSEG	Similar to SPEC-PAIRS (above) except specialty pairings for all grades in the data base can be retrieved and displayed in grade order.	Q,CID = C AND CPREFIX = W AND CGRADE GT Ø AND CACTIVITY GT Ø C,ALL,$GRD = CGRADE / 1Ø C,ALL,3,$RATIO = NRATIO / (NRATIO + MRATIO) S,CGRADE,D,CFROM,CTO P,CFROM,CTO,$GRD,CACTIVITY,$RATIO
NODEACTIVITY	CURSEG	Retrieves all column records representing the input and the output flow of a specific node and year combination for years T_1 through $T_{(N-1)}$, and displays the input arcs with a flow greater than zero, the sum of all input flows, the output arcs with a flow greater than zero, and the sum of the output flows (the query set in the Sample statement listing is for node 11 at T_1, by using the EDIT function within MIRADS the specialty/year combination can be changed).	Q,(CYEAR = Ø AND CTO = 11 AND CACTIVITY GT Ø AND CFROM GT Ø AND (CPREFIX = X OR Y)) OR (CYEAR = 1 AND CFROM = 11 AND CACTIVITY GT Ø) S,CYEAR C,CYEAR,$TOTAL = SUM CACTIVITY P,CPREFIX,CYEAR,CFROM,CTO,CACTIVITY,$TOTAL SP I
PROCUREMENT	CUMSEG	Retrieves all column records for W arcs (whose activity is greater than zero and where the primary and alternate specialty numbers are not the same) from the CPT and MAJ segments, and all row records for the RES_ _ _ _ _ type flow control constraints (whose activity is also greater than zero) from the CPT segment. In addition, retrieves the row records for the node capacity constraints for year T_1 in the LT segment. This data is then input to the PFODSAP.PROCUREMENT program for use in producing the ODSAS Procurement Report.	Q,(CID = C AND CPREFIX = W AND (CGRADE LT 5Ø AND GT 3Ø) AND CACTIVITY GT Ø OR (CID = C AND CPREFIX = W AND CFROM NE *CTO AND CGRADE LT 4Ø AND GT 2Ø AND CACTIVITY GT Ø) OR (ID = R AND PREFIX = N AND YEAR = 1 AND GRADE LT 3Ø) OR (ID = R AND PREFIX = V AND ACTIVITY GT Ø AND GRADE GT 2Ø) P,CFROM,CTO,CACTIVITY @END

TABLE VI-2, Standard Query Sets for Data Base Access (concluded)

Query set name	Applicable data base	Function	Sample statement listing
CUMNODEACT	CUMSEG	Similar to NODEACTIVITY (above) except the cumulative data base records are retrieved. EDITing would also be employed similarly to change the node/year combination.	Q,(CYEAR = Ø AND CTO = 11 AND CACTIVITY GT Ø AND CFROM GT Ø AND (CPREFIX = 7)) OR (CYEAR = 1 AND CFROM = 11 AND CACTIVITY GT Ø) C,ALL,$GRD = CGRADE / 1Ø C,CYEAR,$TOTAL = SUM CACTIVITY S,CYEAR,CGRADE P,CPREFIX,CYEAR,$GRD,CFROM,CTO,CACTIVITY,$TOTAL SP 1
NODEACT-Ø	CURSEG	Retrieves all column records representing the input and the output flow of a specific node at $T_Ø$ only (this is because of unique presence of W arcs as input at $T_Ø$). The display includes in input arcs with a flow greater than zero, the sum of all input flows, the output arcs with a flow greater than zero, and the sum of the output flows (the query set in the Sample statement listing is for node 11; the specialty number can be changed via the EDIT function within MIRADS).	Q,(CYEAR = Ø AND CTO = 11 AND CACTIVITY GT Ø AND CFROM GT Ø AND (CPREFIX = W)) OR (CYEAR = Ø AND CFROM = 11 AND CACTIVITY GT Ø AND (CPREFIX = X OR Y)) S,CPREFIX C,CPREFIX, $TOTAL = SUM CACTIVITY P,CPREFIX,CYEAR,CFROM,CTO,CACTIVITY,$TOTAL SP 1
CUMNODEACTØ	CUMSEG	Retrieves all column records representing the input and the output of a specific node at $T_Ø$ and displays the input arcs with a flow greater than zero, the sum of all input flows, the output arcs with a flow greater than zero, and the sum of the output flows (the EDIT function within MIRADS can be used to change the specialty number).	Q,(CYEAR = Ø AND CTO = 11 AND CACTIVITY GT Ø AND CFROM GT Ø AND (CPREFIX = W)) OR (CYEAR = Ø AND CFROM = 11 AND CACTIVITY GT Ø AND (CPREFIX = Z)) C,CPREFIX,$TOTAL = SUM CACTIVITY C,ALL,$GRD = CGRADE / 1Ø S,CPREFIX,CGRADE P,$GRD,CPREFIX,CYEAR,CFROM,CTO,CACTIVITY,$TOTAL SP 1
CUMGRDSUB	CUMSEG	Retrieves all row records for requirements that were not met by the combined (X+Y) records, computes the requirements not met by officers of the grade required, and prints the results by specialty within year.	Q,ID = R AND PREFIX = N AND ACTIVITY LT *UPLIMIT C,ALL,$GRD-SUB = UPLIMIT - ACTIVITY C,ALL,$GRD = GRADE / 1Ø S,YEAR,GRADE,D,FROM P,YEAR,FROM,$GRD-SUB,$GRD
CUMASSIGN-2	CUMSEG	Retrieves all column records for officers assigned to a particular specialty in a particular year, sorts the records found by specialty within grade, and prints the number assigned for each specialty pair in the required grade and the next lower grade.	Q,CID = C AND CPREFIX = Z AND CYEAR = Ø AND CFROM GT Ø AND CTO = 11 AND CACTIVITY GT Ø S,CGRADE,D,CFROM P,CGRADE,CFROM,CTO,CACTIVITY
CUMASSIGNØ-2	CUMSEG	Retrieves all column records for officers assigned to a particular specialty at $T_Ø$, sorts the records found by specialty within grade, and prints the number assigned for each specialty pair.	Q,CID = C AND CPREFIX = W AND CYEAR = Ø AND CFROM GT Ø AND CTO = 11 AND CACTIVITY GT Ø S,CGRADE,CFROM C,ALL,$PAY-GRADE = CGRADE / 1Ø P,$PAY-GRADE,CFROM,CTO,CACTIVITY

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER VII
DATA BASE RECORD DESCRIPTIONS

1. Purpose. - The purpose of this chapter is to provide descriptions of the types and formats of the records in the two data bases created with MIRADS, in major activity 4 of the processing phase. The two data bases are named, respectively, CURSEG (current segment) and CUMSEG (cumulative results of all segments processed thus far). All other computer disc and tape files are described in Chapter VIII.

2. Types of Records. - There are two types of records in each of the two data bases. The first record type (called row records) contains solution information on the node capacity, control of input constraints, and also flow control constraints for the Y arcs (for the CPT and LT segments only). The second type (called column records) contains information on all the columns, or variables.

3. Data Base Record Formats.* - The number of data fields within each record type differs; there are 9 data fields in a row record in either data base, 15 data fields per column record in the CURSEG data base, and 9 data fields per column record in the CUMSEG data base. The same data field names are used for row and column records in the cumulative data base and the current segment data base. However, the contents are necessarily different because of their functions. One is a cumulative data base; the other contains information on the current segment only. The column records in the CURSEG data base have 6 additional data fields containing information on the arcs with a functional relationship to the column (or variable) which applies. These data are useful for analyzing the current segment solution. In compiling the CUMSEG data base, however, the column record loses its unique identity and thus the functional relationships become useless. Therefore, the data fields for the functional relationships are not part of the column records in the CUMSEG data base.

---

*The row and column record formats are described to the MIRADS system via data base dictionary cards placed in the CUR-DICT (for current segment records) element and the CUM-DICT (for cumulative records) element in the PFODSAP file. (See Chapter IX for detailed description.)

a. Current Segment Data Base Records. - The CURSEG row record format is as shown in Table VII-1, and the CURSEG column record format in Table VII-2.

b. Cumulative Data Base Records

(1) The CUMSEG row record format is as shown in Table VII-3. Since the field names are the same as those used in the CURSEG row records, only the differences in allowable values or definitions are included.

(2) The CUMSEG column record format is as shown in Table VII-4. Since the field names are the same as those used in the CURSEG column records (except for ALT1 through PER2 data fields) only the differences in allowable values or definitions are included.

TABLE VII-1, CURSEG Row Record Format

Field name	Allowable data values	Definition
ID	R	Unique identifier for a row record
PREFIX		Type of constraint
	N	Node capacity constraints (N_ _TREQ for total requirements, N_ _CREQ for unfilled higher grade requirements). See footnote for information on use of GRADE field differentiating CREQ and TREQ records.
	T	Total authorized strength control of input constraint (TOTAUTH constraint)
	W	Authorized strength for specific specialty control of input constraint (WØ_ _UBSG constraints)
	V	Flow control constraint for Y arcs in the CPT and LT segments only (RES_ _ _ _ _constraints)
YEAR	Ø-9	Year to which row record applies
FROM	ØØ-99	Specialty number identifying a node (for PREFIX = V only--"FROM" means primary specialty number of CPT with 8 YOS)
TO	ØØ-99	For PREFIX = V only--alternate specialty number of CPTs with 8 YOS; otherwise the value will be ØØ
GRADE	21-24 31-34 41-44 51-54 61-64	Grade and segment indicator; ((officer grade x 10) + segment number)[a/]
ACTIVITY LOWLIMIT UPLIMIT	12-digit integer number	FMPS solution value for the activity, lower limit and upper limit of the constraint (for PREFIX = N only, the activity of the CREQ constraint with corresponding YEAR and FROM values is subtracted from the activity of the TREQ constraints, since the CREQ constraint refers to requirements for the next higher grade)[a/]

[a/] The segmentation-by-grade and within-grade options introduced the need to identify the segments within a grade, and also to identify the type constraint with a particular grade. First, the unsegmented mode and segment 1 of the segmented-within-grade mode are considered to be a segment indicator of "1" for all records except for the node capacity constraint for the unfilled higher grade requirements (N_ _CREQ). Segment 2 for the segmented-within-grade mode is considered to be segment indicator "2," for all row records except those for the N_ _CREQ constraints. The N_ _CREQ row records are assigned the segment indicator for the N_ _TREQ row record with corresponding values in the YEAR and FROM fields, plus the number 2, thus segment indicators 3, 4. Additionally, the GRADE field value is computed using the grade number corresponding to the grade of the unfilled higher grade requirements (e.g., N_ _CREQ records produced in the LTC segment (grade 5) have GRADE field values of 63 and 64).

TABLE VII-2, CURSEG Column Record Format

Field name	Allowable data values	Definition
CID	C	Unique identifier for a column record
CPREFIX		First letter of variables' name
	W	W arcs in $T'_{\emptyset} - T_{\emptyset}$ interval
	X	X arcs for nonpromotees
	Y	Y arcs for promotees
CYEAR	Ø-9	Year at which assignment is effected; the last year in the projection period is always coded as a "9"
CFROM	ØØ-99	Number of the specialty from which the assignment is effected
CTO	ØØ-99	Number of the specialty to which the assignment is effected
CGRADE	21-24 31-34 41-44 51-54 61-64	Grade and segment indicator (officer grade x 10) + segment number[a/]
CACTIVITY CLOWLIMIT CUPLIMIT	12-digit integer number	FMPS solution value for the activity, lower limit and upper limit of the flow in the variable (arc)
ALT1	5-digit integer number	Number identifies an X arc flow which has a functional relationship to the arc named in the CPREFIX,CYEAR,CFROM,CTO fields
PER1	4-digit integer number	Percent of the X arc flow (multiplied by 1000 to produce an integer value) named in the ALT1 field which equals (at least in part) the value of the variable (arc) named in the CPREFIX, CYEAR,CFROM,CTO fields
PRI	5-digit integer number	Number identifies a W arc flow which has a functional relationship to the arc named in the CPREFIX,CYEAR,CFROM,CTO fields
PER-W	4-digit integer number	Percent of the W arc flow (multiplied by 1000) named in the PRI field which equals (at least in part) the value of the variable (arc) named in the CPREFIX,CYEAR,CFROM,CTO fields
ALT2	5-digit integer number	Number identifies a possible second X arc flow which has a functional relationship to the arc named in the CPREFIX,CYEAR,CFROM,CTO fields
PER2	4-digit integer number	Similar to PER1 except refers to X arc flow named in ALT2

[a/] In a manner similar to the treatment of the row records, segmentation-by-grade and within-grade introduced the need to identify the segments within a grade, and also to identify the type variable with a particular grade. A segment number of 1 applies for either the segment 1 solution in the segmented mode, or the one unsegmented grade solution. Segment number 2 applies to segment 2 solutions only. Since the W and X arcs refer to officers in one grade, and the Y arcs identify officers in the next higher grade, the grade part of the field reflects this difference (e.g., for W and X arcs in the LTC segment the grade part equals 5, whereas for Y, the grade part is a 61. Additionally, the Y arc's segment number is 2 more than the segment number for W and X arcs (3 or 4) so that they can be uniquely identified when updating the cumulative data base.

TABLE VII-3, CUMSEG Row Record Format

Field name	Allowable data values	Definition
ID	a/	a/
		Type of constraint
PREFIX	N	Node capacity constraint for total requirements; the solution values for N_ _ _TREQ and N_ _ _CREQ constraints for corresponding year, specialty and grade (segment indicators 0-4) are combined to reflect results of processing two grade segments (see definition for ACTIVITY LOWLIMIT and UPLIMIT below)
	T	a/
	W	a/
	V	a/
YEAR	a/	a/
FROM	a/	a/
TO	a/	a/
GRADE	20 30 40 50 60	Officer grade x 10; segment indicator (units position) of "0" is indicative of cumulative data base
ACTIVITY LOWLIMIT UPLIMIT	12-digit integer number	Same FMPS solution value for activity, lower limit and upper limit for PREFIX = V, T, and W as in the current segment data base. However for PREFIX = N, these values reflect the solution values obtained after processing two grade segments (to account for promotees filling the unfilled higher grade requirements derived after processing the first grade)

a/ Same as for CURSEG row records

TABLE VII-4, CUMSEG Column Record Format

Field name	Allowable data values	Definition
CID	a/	a/
CPREFIX		First letter of variables' name
	W	W-arcs in $T'_\emptyset - T_\emptyset$ interval
	Z	Z = X+Y arcs for corresponding grade, year and specialties
CYEAR	a/	a/
CFROM	a/	a/
CTO	a/	a/
CACTIVITY CLOWLIMIT CUPLIMIT	a/	Same as for CURSEG except values for *CPREFIX = Z records are the sum of the CPREFIX x* and Y arcs, with corresponding grade, year and specialties, from the current segment data base records

a/ Same as for CURSEG column records.

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER VIII
COMPUTER DISC AND TAPE FILE DESCRIPTIONS

1. Purpose. - The purpose of this chapter is to provide detailed file descriptions of all but the data base disc files utilized in ODSAS. The contents of the data base disc files are described in Chapter VII, and the structure of those files is described in the MIRADS Users Manual (reference 3).

2. Disc and Tape File Naming Convention

a. Qualifier Prefix. - The UNIVAC system requires that disc and tape file names be prefixed by a qualifier consisting of three alphanumeric characters and that this qualifier be separated from the file name by an asterisk. The qualifier for the ODSAS disc and tape files is "CAA."

b. ODSAS Disc and Tape File Names

(1) There are five files (one catalogued runstream file and four program files) whose names begin with the letters PF (for "Program File").

(2) There are also 27 data files, with the names of all but six beginning with the letters OD (for "ODSAS Data"). The six data files whose names do not begin with OD (i.e., INVERSE, MATRIX, UTIL1, UTIL2, SACSTAPEIN, and SACSTAPEOUT) are FMPS work files.

3. Index of ODSAS Disc and Tape File Descriptions. - The remainder of this chapter contains detailed descriptions of the 32 ODSAS disc and tape files. These descriptions are grouped into the two categories in Paragraph 2b, above, and alphabetically within each group. The following index is provided to assist in locating the respective file descriptions.

a. Catalogued Runstream/Program File Descriptions

File name	Page
CAA * PFCAA	VIII-3
CAA * PFFMPS	VIII-5
CAA * PFMIRADS	VIII-7
CAA * PFODSAP	VIII-9
CAA * PFPRINT	VIII-11

b. Data File Descriptions

File name	Page
CAA * INVERSE	VIII-13
CAA * MATRIX	VIII-13
CAA * ODCUMUDØ1	VIII-15
CAA * ODDBSUD1	VIII-17
CAA * ODEQAUDØ1	VIII-19
CAA * ODINPUDØ1	VIII-21
CAA * ODINPUDØ2	VIII-23
CAA * ODOUTUDØ1	VIII-25
CAA * ODPOPUDØ1	VIII-27
CAA * ODRATUDØ1	VIII-29
CAA * ODRECUDØ1	VIII-31
CAA * ODR8SUDØ1	VIII-33
CAA * ODSACUDØ1	VIII-37
CAA * ODSACUDØ2	VIII-39
CAA * ODSAPUDØ3	VIII-41
CAA * ODSAPUDØ4	VIII-43
CAA * ODSAPUDØ7	VIII-45
CAA * ODSAPUD1Ø	VIII-47
CAA * ODSAPUD18	VIII-49
CAA * ODSAVUDØ1	VIII-51
CAA * ODSOLUD1	VIII-53
CAA * ODTURUDØ1	VIII-55
CAA * ODØØ1UD9Ø	VIII-57
CAA * SACSTAPEIN	VIII-59
CAA * SACSTAPEOUT	VIII-61
CAA * UTIL1	VIII-13
CAA * UTIL2	VIII-13

ODSAS
FILE DESCRIPTION

FILE NAME: PFCAA

CATALOGUE OPTIONS: PUBLIC, READ-ONLY

NUMBER OF RECORDS: N/A RECORD SIZE: N/A

RECORD FORMAT: N/A

☐ UNFORMATTED ☐ FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 50 MAXIMUM SIZE - 100

MEDIA:

☐ CARD

☒ DISC, DEVICE TYPE F14

☐ CARD IMAGE ON DISC

☐ TAPE

RECORD FORMAT (IF REQUIRED):
This file contains all the catalogued runstreams used in ODSAS.

ODSAS

FILE DESCRIPTION

FILE NAME: PFFMPS

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: N/A RECORD SIZE: N/A

RECORD FORMAT: N/A

☐ UNFORMATTED ☐ FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 450 MAXIMUM SIZE - 500

MEDIA:

☐ CARD [X] DISC, DEVICE TYPE F60

☐ CARD IMAGE ON DISC ☐ TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
N/A		

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS-MODULE
PFCAA.FMPS-MODULE/CPT-LT
PFCAA.MODIFY
PFCAA.RECOVER

NARRATIVE DESCRIPTION:

This file contains the absolute elements of FMPS.

ODSAS
FILE DESCRIPTION

FILE NAME: PFMIRADS.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: N/A RECORD SIZE: N/A

RECORD FORMAT: N/A

[ ] UNFORMATTED [ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 250 MAXIMUM SIZE - 500

MEDIA:

[ ] CARD [X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC [ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
N/A		

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.MIRADS
PFCAA.IMPLEMENT
PFCAA.ROLLIN
PFCAA.ROLLOUT

NARRATIVE DESCRIPTION:

This file contains all the absolute elements in the MIRADS system.

ODSAS
FILE DESCRIPTION

FILE NAME: PFODSAP

CATALOGUE OPTIONS: PUBLIC, READ-ONLY

NUMBER OF RECORDS: N/A

RECORD SIZE: N/A

RECORD FORMAT:

[ ] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 250

MAXIMUM SIZE - 500

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
N/A		

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS
PFCAA.TOUR
PFCAA.INITIAL
PFCAA.DB-MODULE
PFCAA.LINK-MODULE
PFCAA.MATRIX-MODUL
PFCAA.SEP-MODULE

NARRATIVE DESCRIPTION:

This file contains all the source, relocatable, and absolute elements in the ODSAS system, exclusive of the FMPS and MIRADS elements.

ODSAS
FILE DESCRIPTION

FILE NAME: PFPRINT

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: N/A

RECORD SIZE: 132 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 750

MAXIMUM SIZE - 1,000

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

Each record is a 132-character print line.

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
N/A		

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS	.RUN-COL1	.RUN-LTC2	.RUN-CPT
.TOUR	.RUN-COL2	.RUN-MAJZZ	.RUN-LT
.INITIAL	.RUN-LTCYY	.RUN-MAJ1	
.RUN-COLXX	.RUN-LTC1	.RUN-MAJ2	

NARRATIVE DESCRIPTION:

This file is used to store the breakpointed printed output as unique elements that can be saved and accessed. The element names correspond to the catalogued runstream element name for "SACS," "TOUR," and "INITIAL." That part of the name following the dash in the element names of the "RUN-_ _ _ _" type; e.g., COLXX is the element in PFPRINT that contains the printed output from RUN-COLXX.

ODSAS
FILE DESCRIPTION

FILE NAME: INVERSE, MATRIX, UTIL1, UTIL2

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: Temporary FMPS Work Files RECORD SIZE: Unknown

RECORD FORMAT: Unknown

☐ UNFORMATTED ☐ FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 50 TRKS, (1 POS) MAXIMUM SIZE - 200 TRKS, (4 POS)
*catalogued with POS granularity

MEDIA:

☐ CARD [X] DISC, DEVICE TYPE F60

☐ CARD IMAGE ON DISC ☐ TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFCAA, MEDFMPSABS	X	
or		
PFCAA, LARGE FMPSABS	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS-MODULE
PFCAA.FMPS-MODULE/CPT-LT
PFCAA.MODIFY
PFCAA.RECOVER

NARRATIVE DESCRIPTION:

These are FMPS work files (see FMPS programer's reference manual for functional descriptions).

ODSAS
FILE DESCRIPTION

FILE NAME: ODCUMUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon size of LP problems   RECORD SIZE: 49 characters

RECORD FORMAT:

[ ] UNFORMATTED   [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 100   MAXIMUM SIZE - 500

MEDIA:

[ ] CARD   [ ] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC   [ ] TAPE

RECORD FORMAT (IF REQUIRED):
Same as ODSAPUDØ3, or ODSAPUD18.

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SEPARATE		X
PFODSAP.DBCREATE	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SEP-MODULE
PFCAA.IMPLEMENT

NARRATIVE DESCRIPTION:

This file contains the inactive cumulative data base records of the ODSAS system, i.e., those records for a grade greater than the current grade segment + 1. It can be accessed via MIRADS PFCAA.IMPLEMENT TOTAL which uses ODCUMUDØ1 as input for the CUMSEG data base.

ODSAS
FILE DESCRIPTION

FILE NAME: ODDBSUD1.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: Variable; depends on NYRS, NPREF, NSPEC

RECORD SIZE: 60 DOUBLE PRECISION WORDS

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):*

NORMAL SIZE - 100 TRKS, (2 POS)
*catalogued with POS granularity

MAXIMUM SIZE - 300 TRKS, (5 POS)

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): (see paragraph 2.6.3 in FMPS programers reference manual for full details)

ROW RECORD

(1)	SOLU	TION
(2)	ROWS	
(3)	RN	ILAST
(4)	ROWN	AME1
(5)	ACTI	VITY
(6)	Row Activity Value	
(7)	ROWN	AME1
(8)	LLIM	IT
(9)	Row Lower Limit Value	
(10)	ROWN	AME1
(11)	ULIM	IT
(12)	Row Upper Limit Value	
(13)	ROWN	AME1
(14)	DUAL	ACT
(15)	Dual Activity Value	
(16)	ROWN	AME2
:	(data for row	names)
(60)		

COLUMN RECORD

(1)	SOLU	TION
(2)	COLU	MNS
(3)	RN RN	ILAST
(4)	COLN	AME1
(5)	ACTI	VITY
(6)	Col Activity Value	
(7)	COLN	AME1
(8)	LLIM	IT
(9)	Col Lower Limit Value	
(10)	COLN	AME1
(11)	ULIM	IT
(12)	Col Upper Limit Value	
(13)	ULIM	IT
(14)	DUAL	ACT
(15)	Dual Activity Value	
(16)	COLN	AME2
:	(data	column names)
(60)		

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.DATABASE	X	
PFFMPS.MEDFMPSABS		X
or		
PFFMPS.LARGEFAPSABS		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS.MODULE
PFCAA.DB-MODULE
PFCAA.FMPS.MODULE/CPT-LT
PFCAA.FMPS-CONTROL
PFCAA.FMPS-CPT-LT
PFCAA.RECOVER
PFCAA.FMPS-RECOVER
PFCAA.FMPS-MODIFY
PFCAA.MODIFY

NARRATIVE DESCRIPTION:

Output file of selected FMPS solution data values from current segment used to build data base file for use with information retrieval and display system. The name, activity, lower limit, upper limit, and dual activity values of all requirements constraints (rows) and all variables are in this file.

ODSAS

FILE DESCRIPTION

FILE NAME: ODEQAUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: variable; depends on NYRS, NPREF, NSPED, SEGMENTATION OPTION

RECORD SIZE: 80 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 200

MAXIMUM SIZE - 500

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

Record formats are as described in FMPS programer reference manual for card-image file.

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.MAIN		X
PFCAA.MEDFMPSABS	X	
or		
PFCAA LARGE FMPSABS	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.MATRIX-MODUL
PFCAA.FMPS-MODULE
PFCAA.FMPS-MODULE/CPT -LT

NARRATIVE DESCRIPTION:

This file contains the row, column, right hand side, ranges, bounds, and mask input data descriptions required by the FMPS programs.

ODSAS
FILE DESCRIPTION

FILE NAME: ODINPUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: 28 RECORD SIZE: 80

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 7 MAXIMUM SIZE - 7

MEDIA:

[ ] CARD [X] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC [ ] TAPE

RECORD FORMAT (IF REQUIRED):

Cd 1	I2,2X,I2,3X,2A4,2X,A4,I1;[NSPEC,NYRS,NAME1,NAME2,MODE,JGRADE]	
Cd 2	6(F7.0);[AUTHMX(J),J=1,6]	
Cd 3	11 13;[N,SPECLT(J),J=1,N]	
Cd 4	8(F6.5,',',),F6.4;[ATTHI(1-9)]	
Cd 5	8(F6.4,',',),F6.4;[ATTLO(1-9)]	
Cd 6	8F6.4,',',),F6.4;[PRMT(1-9)]	
Cd 7	3(F6.4,',',),F6.4;[OFLOLO,UFLOLO,OFLOHI,UFLOHI]	
Cd 8	10F6.0[SEG aaa , where aaa = COL, LTC or MAJ]	
Cd 9	8F9.0;[UPBND(K), K=1,50]	
Cd 10-14	10F6.3;[SARRAY(K), K=1,50]	for Grade 6
Cd 15-21	Same as 8-14	for Grade 5
Cd 22-28	Same as 8-14	for Grade 4

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.LINKAGE	X	X
PFODSAP.INPUT1	X	

CATALOGUED RUNSTREAM REFERENCES: PFCAA.START
PFCAA.LINKAGE
PFCAA.MATRIX-MODUL

NARRATIVE DESCRIPTION:

This file contains all the data needed for the matrix generator (less the requirements data) which is needed to produce a FMPS input data file (ODEQAUDØ1.) for any grade segment or subsegment. Parameters for the number of specialties, number of years to project, grade identification, and segmentation option are in this file. The total authorized strength levels by grade, the number and identification of specialties to include in Subsegment 1, attrition rates for promotees and those remaining in grade, as well as the promotion rate, by YOS, are also included. Lastly, the additional information needed to limit the network flow in Subsegment 1 is included. These last data values are present only for the current grade and the field grades not already processed (if any). If a field grade segment is not segmented within grade, then the data are still present, but ignored by the input subroutine.

ODSAS
FILE DESCRIPTION

FILE NAME: ODINPUDØ2

CATALOGUE OPTIONS: PUBLIC, READ-ONLY

NUMBER OF RECORDS: 28 RECORD SIZE: 80 char.

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 7 MAXIMUM SIZE - 7

MEDIA:

[ ] CARD [X] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC [ ] TAPE

RECORD FORMAT (IF REQUIRED):
Same as ODINPUDØ1.

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.INITIAL		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.INITIAL
PFCAA.START

NARRATIVE DESCRIPTION:

This file contains all the data needed for the matrix generator (less the requirements data) to produce a FMPS input data file (ODEQAUDØ1.) for the first, grade 6 segment or subsegment. Parameters for the number of specialties, number of years to project, grade identification, and segmentation option are in this file. The total authorized strength levels by grade, the number and identification of specialties to include in subsegment 1. Attrition rates for promotees and those remaining in grade, as well as the promotion rate, by YOS are also included. Lastly, the additional information needed to limit the network flow in Subsegment 1 is included. This file holds the necessary input parameters for starting the COL segment. It is used to initialize, or re-initialize, the system to the start of processing for grade 6 rather than re-run PFCAA.INITIAL.

ODSAS
FILE DESCRIPTION

FILE NAME: ODOUTUDØ1

CATALOGUE OPTIONS: Not permanently catalogued

NUMBER OF RECORDS: Variable   RECORD SIZE: 45 characters

RECORD FORMAT:

[ ] UNFORMATTED   [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 2,000   MAXIMUM SIZE - 4,000

MEDIA:

[ ] CARD   [X] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC   [ ] TAPE

RECORD FORMAT (IF REQUIRED):

Column	Field Name
1-2	Grade
3-4	Primary specialty
5-6	Alternate specialty
7-19	Filler
20-25	EDATE
26-31	TDATE
32-39	FILLER
41-45	Authorized strength

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SACSPREPRO		X
PFODSAP.SACSCREATE	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS

NARRATIVE DESCRIPTION:

This file contains the SACS data edited for invalid specialty numbers. Some requirements with unofficial OPMS specialty numbers still have to be re-distributed by the PFODSAP.SACSCREATE Program.

ODSAS
FILE DESCRIPTION

FILE NAME: ODPOPUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: 7 RECORD SIZE: 7 31

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 17 MAXIMUM SIZE - 17

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.INITIAL		X
PFODSAP.LINKAGE	X	
PFODSAP.INPUT1	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.INITIAL

PFCAA.LINK-MODULE

NARRATIVE DESCRIPTION: N/A

This file contains the attrition and promotion rates for each grade used in the matrix generator. It also contains the percentage of overfill and underfill of requirements desired by the user for each grade. When processing the captain's segment, the following additional information is needed and is in Record 1: Advanced Entry Specialties, the percentage of captains with less than 8 years of service in each time period, and the time when all captain flows in the network have more than 8 years of service. When processing the lieutenant's segment, the following additional information is needed and is in Record 7: the percentage of lieutenants promoted to captain who have 8 or more years of service in each time period, and the time when those lieutenants promoted to captain will reach their 8th year of service.

Rate Data for all grades per year (9-ATTHI,9-ATTLO,9-PRMT,OFLOHI,UFLOHI,
OFLOLO,UFLOLO)
Row 2 for Grade 2 rates
Row 3 for Grade 3 rates
Row 4 for Grade 4 rates
Row 5 for Grade 5 rates
Row 6 for Grade 6 rates
Additional CPT's data
Row 1 20-AES Codes, 9-CPTRM Rates, 1-NBRAES, 1-ICHG
Additional LT's data
Row 7 20-AES Codes, 9-CPTRM Rates, 1-ICHG

ODSAS
FILE DESCRIPTION

FILE NAME: ODRATUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: 6

RECORD SIZE: 2,500 WORDS (50x50)

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 17

MAXIMUM SIZE - 17

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.TOURATIOS		X
PFODSAP.INPUT1	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.TOUR
PFCAA.MATRIX-MODUL

NARRATIVE DESCRIPTION:

This file contains the utilization ratios and tour lengths of preferred specialty pairings of up to 50 specialties for 50 preferred alternates for Grades O-6 thru O-4.

	Codes
RECORD 6 U.R's for GRADE 6	99 no preference.
RECORD 5 U.R's for GRADE 5	88 preference of combat arm.
RECORD 4 U.R's for GRADE 4	positive number less than 88
RECORD 3 Tour lengths for GRADE 6	indicates utilization ratio for preferred specialty pairing.
RECORD 2 Tour lengths for GRADE 5	(M,M) is location of tour
RECORD 1 Tour lengths for GRADE 4	length of spec M, in records for tour length.

ODSAS
FILE DESCRIPTION

FILE NAME: ODRECUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: unk - FMPS file RECORD SIZE: unk - FMPS file

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):*

NORMAL SIZE - 1 TRK, (1 POS) MAXIMUM SIZE - 5 TRKS, (1 POS)
*catalogued with POS granularity

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):
N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFFMPS.MEDFMPSABS	X	
or		
PFFMPS.LARGEFMPSABS	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS.MODULE
PFCAA.FMPS.MODULE/CPT-LT
PFCAA.RECOVER
PFCAA.MODIFY
PFCAA.POST-OPT

NARRATIVE DESCRIPTION:

This file is used by FMPS in case the computer system crashes. Appendix B to FMPS programers reference manual explains its full function.

ODSAS
FILE DESCRIPTION

FILE NAME: ODR8SUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: 82 RECORD SIZE: 80 characters

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 1 MAXIMUM SIZE - 1

MEDIA:

[ ] CARD [ ] DISC, DEVICE TYPE F14

[X] CARD IMAGE ON DISC [ ] TAPE

RECORD FORMAT (IF REQUIRED):

Cd 1	2I2,2A4: [NSPEC,NYRS,NAME1,NAME2]	
Cd 2	FREE; [AUTHMX(GRADE),GRADE=1,6]	
Cd 3	FREE; [NPRO,(SPECLT(K),K=1,NPRO)]	
Cd 4	FREE; [NBRAES,(AES(K),K=1,NBRAES)]	
Cd 5	FREE; [GRADE,BGNYOS,ENDYOS]	
Cds 6&7	FREE; [POP(GRADE,J),J=BGNYOS,ENDYOS]	Grade 6 Data
Cds 8&9	FREE; [R8WPR(GRADE,J),J=BGNYOS,ENDYOS]	
Cd 10	FREE; [UFLOLO,OFLOLO,UFLOHI,OFLOHI for Grade 6]	
Cds 11-15	Same as cards 5 thru 9	
Cds 16&17	FREE; [R8WOPR(GRADE,J),J=BGNYOS,ENDYOS]	
Cd 18	Same as card 10 for the grade last defined	Grade 5 Data
Cds 19-26	Same as 11-18	Grade 4 Data
Cds 27-34	Same as 11-18	Grade 3 Data
Cd 28	FREE; [GRADE,BGNYOS,ENDYOS]	
Cd 29	FREE; [POP(GRADE,J),J=BGNYOS,ENDYOS]	
Cd 30	FREE; [R8WPR(GRADE,J),J=BGNYOS,ENDYOS]	
Cd 31	FREE; [R8WOPR(GRADE,J),J-BGNYOS,ENDYOS]	Grade 2 Data
Cd 32	FREE; [UFLOLO,DFLOLO,UFLOHI,OFLOHI for Grade 2]	
Cd 33*	FREE; LTR8W(GRADE,J),J-ENDYOS,ENDYOS+NYRS]	

RECORD FORMAT (continued): ODR8SUDØ1

```
Cd 34*        FREE; [LTR8WO(GRADE,J),J=ENDYOS, ENDYOS+NYRS]           Grade 2 Data
Cds 35-41     Same as 28-34                                           Grade 1 Data
Cd 42         FREE; [KGRADE,SEGCOL,LGRADE,SEGLTC,MGRADE,SEGMAJ]
Cds 43-52     FREE; [spec #1, SARRAY(1), ... spec #5, SARRAY(5)
                     spec #6, SARRAY(6), ... spec *10,SARRAY(10)
                       .          .          .
                       .          .          .
                       .          .          .                       Grade 6 Data
                     spec #46, SARRAY(46)... spec #50, SARRAY(50)]
Cds 53-62     FREE; [spec #1, UPBND(1), ... spec #5, UPBND(5)
                       .          .          .
                       .          .          .
                       .          .          .
                     spec #N46, UPBND(46), ... spec #50, UPBND(50)]
Cds 63-72     Same as 43-62                                           for Grade 5
Cds 73-82     Same as 43-62                                           for Grade 4
```

---

*Additional Attrition Rates for Lieutenants

FILE NAME: ODR8SUDØ1 (continued)

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.INITIAL	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.INITIAL

NARRATIVE DESCRIPTION:

This is one of the two input files with user-supplied data (the other one is ODRATUDØ1). A third input file, SACS tape file, is furnished by USAMSSA. Parameter values for the number of specialties, number of years to project, problem name, total authorized amounts by grade, identification of specialties to include in subsegment 1 for the field grade segments, and identification of advanced entry specialties are input via this file. Populations and two sets of rates (attrition rates to include losses caused by promotion to next higher grade and attrition rates that exclude losses caused by promotion to next higher grade) are input for the years of service to be represented for grades 0-6 thru 0-1. Parameter values that determine which, if any, field grade segments will be processed in two subsegments are also input. If any of the field grade segments are to be processed in two subsegments, then additional data is input used to limit the amount of flows into the specialties in subsegment 1.

ODSAS

FILE DESCRIPTION

FILE NAME: ODSACUDØ1.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: 1

RECORD SIZE: 6,600 (11,6,100) words

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 7

MAXIMUM SIZE - 7

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.LINKAGE	X	X
PFODSAP.INPUT1	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.START
PFCAA.LINK-MODULE
PFCAA.MATRIX-MODUL

NARRATIVE DESCRIPTION:

Contains specialty requirements created from SACS requirement data tape by year, grade, and actual specialty number.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSACUDØ2

CATALOGUE OPTIONS: PUBLIC, READ-ONLY

NUMBER OF RECORDS: 1 RECORD SIZE: 6,600 (11, 6, 100) Words

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 7 MAXIMUM SIZE - 7

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.INITIAL		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.INITIAL
PFCAA.START

NARRATIVE DESCRIPTION:

Contains specialty requirements created from SACS requirements data tape by year, grade, and actual specialty number. This file holds the requirements values at the start of processing. It is used to initialize, or re-initialize the system to the start of processing for grade 6 rather than rerun PFCAA.SACS.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAPUDØ3

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon LP problem size RECORD SIZE: 49 characters

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 50 MAXIMUM SIZE - 200

MEDIA:

[ ] CARD [ ] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC [ ] TAPE

RECORD FORMAT (IF REQUIRED):

Column	Name	
1	Identity	
2	Prefix	
3	Year	
4-5	From Specialty	
6-7	To Specialty	
8-9	Segment ID	
10-21	Activity	
22-33	Lower Limit	
34-45	Upper Limit	
46	Mratio	Ø filled if Identity is "R" - Row record
47	Nratio	
48	Tour Length M	
49	Tour Length N	

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSP.DATABASE		X
PFODSP.DBCREATE	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.DB-MODULE
PFCAA.IMPLEMENT
PFCAA.LINK-MODULE
PFCAA.SEP-MODULE

NARRATIVE DESCRIPTION:

The file contains the cumulative file of active segments. It is created by PFODSAP.DATABASE and then analyzed by the user with the MIRADS system. Upon acceptance of the segment, this file is copied into ODSAPUD18, which becomes the record file. This file is purged of inactive grade records during the PFCAA.SEP-MODULE run. It is the input from which MIRADS constructs the CUMSEG data base.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAPUDØ4

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon size of LP problem

RECORD SIZE: 38 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 10

MAXIMUM SIZE - 100

MEDIA:

[ ] CARD

[ ] DISC, DEVICE TYPE F60

[X] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

Column	Format	Name
1	A1	Prefix
2	I1	Year
3-4	I2	From Specialty (M)
5-6	I2	To Specialty (N)
7	IX	Blank
8	I1	Mratio
9	I1	Nratio
10	I1	Tour length-M
11	I1	Tour length-N
12-16	I5	First ALT Source
17-20	I4	Percent of 1st ALT Source
21-25	I5	Primary Source
26-29	I4	Percent of Primary Source
30-34	I5	2d ALT Source
35-38	I4	Percent of 2d Alternate

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SORTW	X	X
PFODSAP.SORTXY		X
PFODSAP.COMBIN	X	
PFODSAP.LOCOLS		X
PFODSAP.RES2GR		X
PFODSAP.LOCOLC		X
PFODSAP.LOCOLL		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.MATRIX-MODUL
PFCAA.DB-MODULE

NARRATIVE DESCRIPTION:

This file is created by the matrix generator to be combined with the data base solution file generated by FMPS (ODDBSUD1.) to create the data base. The data are drawn from several routines within the matrix generator and then sorted before combining to produce the data base file for the segment (ODSAPUDØ7).

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAPUD07

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon LP problem size

RECORD SIZE: 76 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 150

MAXIMUM SIZE - 500

MEDIA:

[ ] CARD

[ ] DISC, DEVICE TYPE F14

[X] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

Column	Name
1	Identity
2	Prefix
3	Year
4-5	From Specialty
6-7	To Specialty
8-9	Segment ID
10-21	Activity
22-33	Lower Limit
34-45	Upper Limit
46	Mratio
47	Nratio
48	Tour Length-M
49	Tour Length-N
50-54	First Alternate Source
55-58	Percent of First Alternate Source
59-63	Primary Source
64-67	Percent of Primary Source
68-72	Second Alternate Source
73-76	Percent of Second Alternate Source

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.COMBIN		X
PFODSAP.RECORD	X	
PFODSAP.DBCREATE	X	
PFODSAP.DATABASE		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.DB-MODULE
PFCAA.IMPLEMENT

NARRATIVE DESCRIPTION:

This file contains the solution information from the FMPS activity pertaining to the Grade/Segment just finished and the data items for each row and column produced in the matrix generator activity. It contains all the input from which MIRADS constructs the CURSEG data base.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAPUD1Ø

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon amount of SACS data

RECORD SIZE: 6,600 words (11,6,100)

RECORD FORMAT:

[X] UNFORMATTED

[ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 5

MAXIMUM SIZE - 10

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SACSCREATE		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS

NARRATIVE DESCRIPTION:

This file contains the SACS requirements by specialty and grade as derived from latest SACS tape. Once the user is satisfied with the new SACS data, then this file is copied to ODSACUDØ2 (after removing the read-only option).

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAPUD18

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends upon LP problem size

RECORD SIZE: 49 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 200

MAXIMUM SIZE - 500

MEDIA:

[ ] CARD

[ ] DISC, DEVICE TYPE F14

[X] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):

Column	Name	
1	Identity	
2	Prefix	
3	Year	
4-5	From Specialty	
6-7	To Specialty	
8-9	Segment ID	
10-21	Activity	
22-33	Lower Limit	
34-45	Upper Limit	
46	Mratio	Ø filled if Identity is "R" - Row record
47	Nratio	Ø filled if Identity is "R" - Row record
48	Tour Length-M	Ø filled if Identity is "R" - Row record
49	Tour Length-N	Ø filled if Identity is "R" - Row record

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SEPARATE	X	X
PFODSAP.RECORD	X	
PFODSAP.DBCREATE	X	
PFODSAP.DATABASE	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.DB-MODULE
PFCAA.SEP-MODULE
PFCAA.IMPLEMENT
PFCAA.LINK-MODULE

NARRATIVE DESCRIPTION:

The file contains those cumulative (active) records (for grades that have not had all possible input considered) in the ODSAS system. Upon completion of a segment and the solution approved by the user, the file is updated by copying ODSAPUDØ3 into the file and then separating out all records pertaining to records whose grade is greater than (current grade plus one); e.g., upon approval of the last LTC segment (grade 5), all COL records (grade 6) would be separated. This file is the permanent cumulative data base, whereas ODSAPUDØ3 is the temporary cumulative data base.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSAVUDØ1.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: Variable   RECORD SIZE: Unk.   FMPS file

RECORD FORMAT:

[X] UNFORMATTED   [ ] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS)*:

NORMAL SIZE - 200 TRKS (4 POS)   MAXIMUM SIZE - 500 TRKS (8 POS)

*Catalogued with POS granularity

MEDIA:

[ ] CARD   [X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC   [ ] TAPE

RECORD FORMAT (IF REQUIRED): N/A

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS-CONTROL
PFCAA.FMPS-MODULE
PFCAA.RECOVER
PFCAA.FMPS-MODULE/CPT-LT
PFCAA.FMPS-CPT-LT
PFCAA.FMPS-MODIFY
PFCAA.FMPS-RECOVER
PFCAA.FMPS-POST
PFCAA.POST-OPT

NARRATIVE DESCRIPTION:

Save file for saving basis and restarting FMPS solutions. Written to when procedure 'Call Save' is specified and read from when procedure 'Call Restore' is specified as FMPS control statements.

ODSAS
FILE DESCRIPTION

FILE NAME: ODSOLUD1.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE; depends on NYRS, NPREF, NSPEC   RECORD SIZE: 60 DOUBLE PRECISION WORDS

RECORD FORMAT:

[ ] UNFORMATTED   [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):*

NORMAL SIZE - 10 TRKS, (1 POS)   MAXIMUM SIZE - 50 TRKS, (1 POS)

*Catalogued with POS granularity

MEDIA:

[ ] CARD   [X] DISC, DEVICE TYPE F60

[ ] CARD IMAGE ON DISC   [ ] TAPE

RECORD FORMAT (IF REQUIRED): (see para 2.6.3 in FMPS programmers reference manual for full details)

ROW RECORD

(1)	SOLU	TION
(2)	ROWS	
(3)	RN	ILAST
(4)	ROWN	AME1
(5)	AT	
(6)	AA	
(7)	ROWN	AME1
(8)	ACTI	VITY
(9)	Row Activity Value	
(10)	ROWN	AME1
(11)	LLIM	IT
(12)	Lower Limit Value	
(13)	ROWN	AME1
(14)	ULIM	IT
(15)	Upper Limit Value	
(16)	ROWN	AME2
⋮	⋮	⋮
(60)	(data for row names)	

COLUMN RECORD

(1)	SOLU	TION
(2)	COLU	MNS
(3)	RN	ILAST
(4)	COLN	AME1
(5)	AT	
(6)	AA	
(7)	COLN	AME1
(8)	ACTI	VITY
(9)	Column Activity Value	
(10)	COLN	AME1
(11)	LLIM	IT
(12)	Column Lower Limit Value	
(13)	COLN	AME1
(14)	ULIM	IT
(15)	Column Upper Limit Value	
(16)	COLN	AME2
⋮	⋮	⋮
(60)	(data for column names)	

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.LINKAGE	X	
PFFMPS.MEDFMPSABS or PFFMPS.LARGEFMPSABS		X

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.FMPS-MODULE
PFCAA.LINK-MODULE
PFCAA.FMPS-MODULE/CPT-LT
PFCAA.FMPS-CONTROL
PFCAA.RECOVER
PFCAA.FMPS-MODIFY
PFCAA.FMPS-CPT-LT
PFCAA.FMPS-RECOVER

NARRATIVE DESCRIPTION:

Output file for selected parts of FMPS solution, to input to segment linkage major activity. Contains the name, activity code, activity, lower limit, and upper limit values for all requirements constraints (rows) and all arcs (variables) named XN--. These data are used to compute the unfilled requirements resulting from the last segment solution.

ODSAS
FILE DESCRIPTION

FILE NAME: ODTURUDØ1

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: Variable; 2 parameter cards and approx. 300 preferred specialty pairings   RECORD SIZE:

RECORD FORMAT:

[ ] UNFORMATTED   [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - 2   MAXIMUM SIZE - 128

MEDIA:

[ ] CARD   [ ] DISC, DEVICE TYPE F14

[X] CARD IMAGE ON DISC   [ ] TAPE

RECORD FORMAT (IF REQUIRED):

Cd 1	A6;	[MODE]
Cd 2	FREE;	[NBRPRO, NPROB(J),J=1, NBRPRO]
Cd 3-N	2I2, 3(F5.3,I2);	[IPR, IALT, (UTIL(K), TOUR(K), K-6,4)]

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.TOURATIOS	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.TOUR

NARRATIVE DESCRIPTION:

The first card image is either "CREATE" or "UPDATE." If "CREATE," then new ODRATUDØ1 is to be created; otherwise, "UPDATE" implies ODRATUDØ1 is to be updated with following data cards. The second card identifies those specialties that cannot be alternates. The rest of the file contains the preferences for specialty pairings. Along with each preference is the utilization ratio and the tour length of the primary specialty in each grade 6 through 4. The tour length is multiplied by 10 for internal program logic; i.e., a tour length of 3 is entered as 30.

ODSAS
FILE DESCRIPTION

FILE NAME: ODØØ1UD9Ø

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: VARIABLE

RECORD SIZE: 132 characters

RECORD FORMAT:

[ ] UNFORMATTED

[X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE (# OF TRKS):

NORMAL SIZE - Variable

MAXIMUM SIZE - 1,000

MEDIA:

[ ] CARD

[X] DISC, DEVICE TYPE F14

[ ] CARD IMAGE ON DISC

[ ] TAPE

RECORD FORMAT (IF REQUIRED):
Each record is a 132-character print line

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
N/A		

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS	.RUN-COL1	.RUN-LTC2	.RUN-CPT
.TOUR	.RUN-COL2	.RUN-MAJZZ	.RUN-LT
.INITIAL	.RUN-LTCYY	.RUN-MAJ1	.MODIFY
.RUN-COLXX	.RUN-LTC1	.RUN-MAJ2	

NARRATIVE DESCRIPTION:

This file is the system breakpoint file. Upon normal termination of a runstream, this file is copied to PFPRINT as an element (@COPY,I) except for the "MODIFY" runstream, which is SYM'd to the printer.

ODSAS
FILE DESCRIPTION

FILE NAME: SACSTAPEIN. (called SACS "A" Tape; Personnel and Equipment Detail Record File)

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: Variable. RECORD SIZE: 76 characters, blocked 95

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE:

NORMAL SIZE - 1 reel of tape MAXIMUM SIZE - 2 reels of tape

MEDIA:

[ ] CARD [ ] DISC, DEVICE TYPE ________

[ ] CARD IMAGE ON DISC [X] TAPE 800 BPI, EBCDIC

RECORD FORMAT (IF REQUIRED):

Characters	Field Name
1-6	FILLER
7-12	EDATE
13-18	TDATE
19	FILLER
20-21	GRADE (e.g., 06)
22-40	FILLER
41-42	PRIMARY SPECIALTY
43-44	ALTERNATE SPECIALTY
45-53	FILLER
54-58	AUTHORIZED STRENGTH
59-76	FILLER

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SACSEXTRACT (CARD DECKS)	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACSEXTRACT (copy of card deck runstream used on System 3 for conversion to 16ØØ BPI, FIELDATA for use on System 2).

NARRATIVE DESCRIPTION:

This is the PERSACS tape file generated by USAMSSA, from which the positional requirements for ODSAS are drawn.

ODSAS
FILE DESCRIPTION

FILE NAME: SACSTAPEOUT.

CATALOGUE OPTIONS: PUBLIC

NUMBER OF RECORDS: *Variable* RECORD SIZE: *45 characters*

RECORD FORMAT:

[ ] UNFORMATTED [X] FORMATTED (IF SO, THEN SPECIFY BELOW)

FILE SIZE:

NORMAL SIZE - 1 reel *MAXIMUM SIZE - 1 reel*

MEDIA:

[ ] CARD [ ] DISC, DEVICE TYPE ________

[ ] CARD IMAGE ON DISC [X] TAPE 1,600 BPI, FIELDATA

RECORD FORMAT (IF REQUIRED):

Characters	Field Name
1	FILLER
2	GRADE
3-4	Primary specialty
5-6	Alternate specialty
7-19	Filler
20-25	EDATE
26-31	TDATE
32-40	FILLER
41-45	AUTHORIZED STRENGTH
46-80	FILLER

INTERACTIONS WITH PROGRAMS:

PROGRAM NAME	INPUT	OUTPUT
PFODSAP.SACSEXTRACT		X
PFODSAP.SACSPREPRO	X	

CATALOGUED RUNSTREAM REFERENCES:

PFCAA.SACS

PFCAA.SACSEXTRACT (copy of card deck runstream used on System 3 for conversion to 16ØØ BPI, FIELDATA on System 2)

NARRATIVE DESCRIPTION:

This tape contains only the PERSACS data that pertain to officer grades 0-1 through 0-6.

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER IX
CATALOGUED RUNSTREAMS

1. Purpose. - This chapter describes the format and provides listings of the catalogued runstreams used in ODSAS.

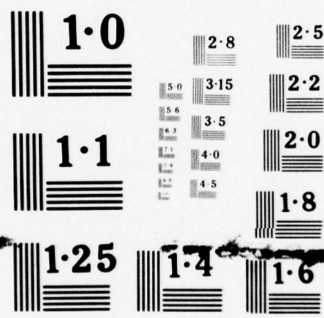
a. Hierarchical Structure of Runstreams. - As described in Chapter VI, Operations Guide, the user commands to execute any of the runstreams are either "@START" or "@ADD,L," followed by a file name and an element name. The element named in the user's command contains a set of EXEC 8 commands that may themselves be references to catalogued runstreams. Thus, a hierarchy of EXEC 8 commands is used to simplify the operation of the system for the user, and to define and separate the subsets of the EXEC 8 commands that perform specific functions. For example, the command--@START PFCAA.RUN-COLXX, which starts the COL segment--references the element named "RUN-COLXX" in a file named PFCAA. The element "RUN-COLXX" contains the following:

```
@RUN,/TP OD006X,090112.CAA,500,1000
@TYPE 2,BA
@PRT,I
@ASG,A OD001UD90.
@BRKPT PRINT$/OD001UD90
@HDG,P COLONEL UNSEGMENTED  UNCL                                 UNCLASSIFIED
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA,IMPLEMENT
@END
@ASG,A ODINPUD01.
@ASG,A PFCAA.
@ED,I SEGMENT
61
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE OD001UD90.
@ASG,A OD001UD90.
@ASG,A PFPRINT.
@COPY,I OD001UD90.,PFPRINT.COLXX
@SYM,SU OD001UD90.,,PR
@FIN
```

The statements with the arrows in left margin are commands to execute other catalogued runstreams. For instance, the first statement with an arrow (@ADD,L PFCAA.DB-MODULE) refers to a

AD-A040 832 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 5/9
OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS) DOCUMENTATION.(U)
MAY 77 J D THOMAS, G E ARMSTRONG
UNCLASSIFIED CAA-D-77-1 NL

3 OF 5
ADA
040832



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

catalogued runstream in the element named "DB-MODULE", listed below:

```
@ASG,A  ODDBSUD1.                                   . BEGIN DATA BASE MODULE
@ASG,A  ODSAPUD03
@ASG,A  ODSAPUD04
@ASG,A  ODSAPUD07
@ASG,A  ODSAPUD18
@DELETE,C 10TEMP.
@PRT,I
@PRT,F ODSOLUD1.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P 10TEMP.,F2/640/TRK/3000
@ASG,A 10TEMP.
@USE 10.,10TEMP.
@USE 11,ODDBSUD1
@USE  3,ODSAPUD03
@USE  4,ODSAPUD04
@USE  7,ODSAPUD07
@USE 18,ODSAPUD18
@XQT PFODSAP.DATABASE
@ADD,LP SEGMENT
@PMD,E
@FREE 3
@FREE 4
@FREE 7
@FREE 10
@FREE 11
@FREE 18
@ASG,A ODSAPUD03.
@ASG,T 20.
@ASG,T 21.
@ASG,T 22.,F60///1000
@ASG,A ODINPUD01.
@USE 16.,ODINPUD01.
                         . LOCATE BEGINNING OF COLUMN RECORDS
                 . PUT ROW RECORDS IN FILE 20.
                         . GO TO END OF FILE
                . PUT COLUMN RECORDS IN FILE 21.
                         . LEAVE ODSAPUD03 UNCHANGED
@ED,U ODSAPUD03.
F CW
SPLIT 20.
LAST
SPLIT! 21.
OMIT
@XQT PFODSAP.DB-CORRECT
@FREE 16.
@FREE 20.
@FREE 21.
@TEST TE/6/T3              . SKIP NEXT STMT IF GRADE = 6
@COPY 22.,ODSAPUD03.
@FREE 22.
```

This is an example of the lowest level of the hierarchy, and here EXEC 8 commands are first encountered that specify appropriate files be assigned and/or released, and that ODSAS programs be executed.

b. Index of Catalogued Runstreams. - Listings of all the catalogued runstreams in this chapter, and a function statement for each, are provided in the remainder of the chapter. Listings are grouped in three categories--initialization phase runstreams, processing phase runstreams, and ancillary runstreams. The following alphabetical index is provided to assist in locating the respective runstreams

(1) Initialization Phase Runstreams	Page
PFCAA.INITIAL	IX-7
PFCAA.SACS	IX-6
PFCAA.SACSEXTRACT	IX-4
PFCAA.TOUR	IX-7
PFCAA.START	IX-7

(2) Processing Phase Runstreams	
PFCAA.DB-MODULE	IX-18
PFCAA.FMPS-CONTROL	IX-19
PFCAA.FMPS-CONTROL/CPT-LT	IX-22
PFCAA.FMPS-MODULE	IX-16
PFCAA.FMPS-MODULE/CPT-LT	IX-17
PFCAA.FREE	IX-15
PFCAA.IMPLEMENT	IX-18
PFCAA.LINK-MODULE	IX-14
PFCAA.MATRIX-MODUL	IX-15
PFCAA.MESSAGE	IX-28
PFCAA.MESSAGE-ONE	IX-28
PFCAA.MESSAGE-TWO	IX-28
PFCAA.PROCURE	IX-27
PFCAA.RUN-COLXX	IX-9
PFCAA.RUN-COL1	IX-11
PFCAA.RUN-COL2	IX-12
PFCAA.RUN-CPT	IX-10
PFCAA.RUN-LT	IX-11
PFCAA.RUN-LTCYY	IX-10
PFCAA.RUN-LTC1	IX-12
PFCAA.RUN-LTC2	IX-13
PFCAA.RUN-MAJZZ	IX-10
PFCAA.RUN-MAJ1	IX-13
PFCAA.RUN-MAJ2	IX-14
PFCAA.SAVECOPY/TOTAPE	IX-27
PFCAA.SEP-MODULE/1	IX-26
PFCAA.SEP-MODULE/2	IX-25
PFCAA.SEP-MODULE/3	IX-25
PFCAA.SEP-MODULE/4	IX-24

(3) Ancillary Runstreams Page

PFCAA.FMPS-MODIFY	IX-31
PFCAA.FMPS-RECOVER	IX-35
PFCAA.IMPLEMENT/TOTAL	IX-37
PFCAA.MODIFY	IX-29
PFCAA.MODIFY/CARDS	IX-30
PFCAA.MODIFY/CARDS-CPT-LT	IX-31
PFCAA.RECOVER	IX-33
PFCAA.RECOVER/CPT-LT	IX-34
PFCAA.ROLLIN	IX-38
PFCAA.ROLLOUT	IX-38
PFCAA.SAVECOPY/TODISC	IX-37
PFCAA.STD-QUERY/CUMSEG	IX-40
PFCAA.STD-QUERY/CURSEG	IX-39
PFCAA.UPDATE	IX-29
PFODSAP.CUM-DICT	IX-41
PFODSAP.CUR-DICT	IX-41
PFODSAP.MAP/DATABASE	IX-44
PFODSAP.MAP/DBLOAD	IX-44
PFODSAP.MAP/INITIAL	IX-42
PFODSAP.MAP/LINKAGE	IX-44
PFODSAP.MAP/MATRIX	IX-43
PFODSAP.MAP/PROCUREMENT	IX-45
PFODSAP.MAP/SACS	IX-42
PFODSAP.MAP/SEPARATE	IX-44
PFODSAP.MAP/TOURATIOS	IX-43

2. Initialization Phase Runstreams. - The name, function and listing of the four initialization phase runstreams are as follows:

a. PFCAA.SACSEXTRACT. - Extracts requirements data from USAMSSA tape for use as input on System 2. Following is a copy of the runstream, including the FORTRAN source program, used to extract the specialty requirements from the SACS tape produced at USAMSSA on an IBM computer. A translation of the IBM generated tape is required from 8ØØ BPI, EBCDIC to 16ØØ BPI, FIELDATA and must be done on a computer system with that translation capability. In July 1976, this translation capability existed only on System 3 at MILPERCEN. The original runstream is in card deck form and is submitted as a batch job, since file space is not catalogued and saved on System 3.

PFCAA.SACSEXTRACT Runstream:

```
@HDG,P  CPT VON LOEWENFELDT 1N49 PSS-A  @* CAA TEST RUN **
@ASG,T UDC12UT01.,16N,OUTPUT
@ASG,T  11.,F///BOD
@FOR,IS    .SACSEXTRACT
       IMPLICIT INTEGER (A-Z)
       DIMENSION IN(1203),OUT(13),IASG(7),NTAPEX(6)
        DATA IASG/42H@ASG,T      10,16D///EBCDIC//6,
       COMMON /OPTION/IWRITE
C
C       @ASG,THIE  TAPEIN.,8C,5XXXR
C       @ASG,T        TAPEOUT.,8C9,SAVEW
C       @USE 10,TAPEIN.
C       @USE 11,TAPEOUT.
C
       READ(5,930)NTAPE
       WRITE(6,932)NTAPE
       READ(5,931)(NTAPEX(I),I = 1,NTAPE)
       WRITE(6,933)(NTAPEX(I),I= 1,NTAPE)
       GO TO 2
C
1       CALL NTRAN(10,2,1203,IN,L,22)
        IF(L.EQ.-2) GO TO 2
       IZ = 0
       ICTBLK= ICTBLK + 95
       DO 100 IJ = 1,95,3
       IZ = IZ + 1
       IJ1= (IZ-1)*38  + 1
        DECODE(76,911,IN(IJ1))OUT
       CALL FIX(OUT)
       IJ2=IJ1+12
        DECODE(80,912,IN( IJ2))X,OUT
       CALL FIX(OUT)
       IF((IJ+2).GT.95) GO TO 1
       IJ3=IJ1+25
        DECODE(78,913,IN( IJ3))X,OUT
       CALL FIX(OUT)
   911 FORMAT(12A6,A4)
   912 FORMAT(A4,12A6,A4)
   913 FORMAT (A2,12A6,A4)
930    FORMAT()
931    FORMAT(6A6)
932    FORMAT(' NUMBER OF TAPES BEING READ EQUALS ',I4)
933    FORMAT(' TAPE NUMBERS ARE ',6(A6,',',2X))
100    CONTINUE
        GO TO 1
2       CONTINUE
       ITAPE = ITAPE + 1
       IF (ITAPE.GT.NTAPE)GO TO 30
       IF(ITAPE.GT.1)  CALL ERTRAN(6,'@FREE 10. . ')
       IASG(6) = NTAPEX(ITAPE)
       CALL ERTRAN(6,IASG)
       GO TO 1
30      ENDFILE 11
        ENDFILE 11
       WRITE(6,940)ICTBLK,IWRITE

 940   FORMAT(' NUMBER OF RECORDS PROCESSED EQUALS ',I8,/
      * ' NUMBER OF RECORDS EXTRACTED EQUALS ',I8)
        STOP
       SUBROUTINE FIX(IOUT)
       COMMON /OPTION/IWRITE
       DIMENSION IOUT(13)
       FLD(0,6,ICOMP)=FLD(30,6,IOUT(12))
       IF(ICOMP.NE.'1@@@@@')RETURN
C
       FLD(0,6,IOFF)=FLD(6,6,IOUT(4))
       IF(IOFF.NE.'O@@@@@')RETURN     @ TEST FOR OFFICER RECORD
C
       IGRADE = FLD(12,6,IOUT(4))      @ EXTRACT GRADE
       IF(IGRADE.GT.84)RETURN          @ CONSIDER COL-LT ONLY
C
       IEDATE = FLD(0,36,IOUT(2))      @ START DATE
       ITDATE = FLD(0,36,IOUT(3))      @ TERMINAL DATE
```

PFCAA.SACSEXTRACT Runstream (continued)

```
C
      IPRMY = FLD(18,12,IOUT(4))      @ EXTRACT PRIMARY SPECIALTY
      IKEY = FLD(30,6,IOUT(4))      @ EXTRACT PRIMARY SPECIALTY KEY
      IF(IKEY.GT.47) IPRMY = FLD(24,12,IOUT(7))
      IAUTH = FLD(0,24,IOUT(10))      @ AUTHORIZED QUANTITY
      FLD(6,6,IAUTH) = FLD(30,6,IOUT(9))      @ AUTHORIZED QTY
      IWRITE = IWRITE + 1
      WRITE(11,902)IGRADE,IPRMY,IEDATE,ITDATE,IAUTH
902   FORMAT(1X,R1,R2,15X,A6,A6,9X,R5)
901   FORMAT(1X,12A6,A4)
      RETURN
      END
@MAP,S   ,ABS
@XQT
2
HF058 HF043
@PMD,LE
@COPY,GM 11.,ODC12UT01.
@EOF
```

b. PFCAA.SACS. - Creates the master specialty requirements file, ODSACUDØ2, and the specialty requirements report.

```
@RUN,J/RTP 000015,090112,CAA,300,500
@TYPE 2,8A
@DELETE,C OD001UD90.
@PRT,I
@ASG,UP OD001UD90.
@BRKPT PRINT$/OD001UD90
@HDG,P  SACS FILE CREATION FOR ODSAS     UNCLASSIFIED
@ASG,T TAPE.,16N,10363
@DELETE,C TEMPDISC.
@ASG,UP TEMPDISC.,F/9/POS/60       . TEMP DISC FILR
@USE 10.,TEMPDISC.
@COPY,G TAPE.,10.
@REWIND TAPE.
@FREE TAPE.
@DELETE,C ODOUTUD01.
@PRT,I
@ASG,UP  ODOUTUD01.,F//POS/300
@USE 11.,ODOUTUD01.
@XQT  PFODSAP.SACSPREPRO
@PMD,EL
@FREE,A 10.
@FREE,A 11.
@PRT,F ODOUTUD01.
@USE 10.,ODOUTUD01.
@ASG,A ODSAPUD10.
@USE 11.,ODSAPUD10.
@PRT,I
@XQT  PFODSAP.SACSCREATE
770930
@PMD,E
@PRT,F ODSAPUD10.
@PRT,I
@BRKPT PRINT$
@FREE OD001UD90.
@SYM,U OD001UD90.,,PR
@ASG,A OD001UD90.
@ASG,A PFPRINT.
@COPY,I OD001UD90.,PFPRINT.PRINT
@END
```

c. PFCAA.TOUR. - Creates the specialty preferences file, ODTURUDØ1.

```
@RUN,/TP ODOO1,090112,CAA,100,200
@TYPE 2,8A
@HDG,P  PFCAA-TOUR    ODSAS  CALL CPT TOFFLER 325-8152
@DELETE,C ODOO1UD90.
@ASG,A ODOO1UD90.
@BRKPT PRINT$/ODOO1UD90
@HDG,P          PFCAA/TOUR     ODSAS
@ASG,A ODTURUD01.
@ASG,A ODRATUD01.
@USE 10.,ODRATUD01.
@ASG,A PFODSAP.
@XQT PFODSAP.TOURATIOABS
CREATE
9,11,12,13,14
@ADD,P ODTURUD01.
@DATA,L ODTURUD01.
@END
@BRKPT PRINT$
@FREE ODOO1UD90.
@SYM,SU ODOO1UD90.,,PR
@FIN
```

d. PFCAA.INITIAL. - Creates the master input parameters file, ODINPUDØ2, the attrition and promotion rates file, ODPOPUDØ1, and the attrition and promotion rates report.

```
@RUN,/TP    ODOO1,090112,CAA,20,500
@TYPE 2,8A
@DELETE,C ODOO1UD95.
@ASG,UP ODOO1UD95.
@BRKPT PRINT$/ODOO1UD95
@HDG,P UNCLASSIFIED (PFCAA/INITIAL)  CALL CPT TOFFLER  325-8152
@ASG,A PFODSAP.
@ASG,A ODR8SUD01.
@DATA,L ODR8SUD01.
@END
@ASG,A    ODPOPUD01.
@USE 10.,ODPOPUD01.
@ASG,A ODINPUD02.
@USE 11.,ODINPUD02.
@CHG,Z ODINPUD02.            . REMOVE R/O MODE
@XQT PFODSAP.INITIAL
@ADD,P ODR8SUD01.
@FREE,X ODINPUD02.
@DATA,L ODINPUD02.
@END
@CHG,V ODINPUD02.            . SET R/O MODE
@BRKPT PRINT$
@FREE ODOO1UD95.
@SYM,SU ODOO1UD95.,,PR
@END
```

e. PFCAA.START. - Copies the master input parameters file (ODINPUDØ2) to the input parameters file used in the system (ODINPUDØ1), and the master specialty requirements file (ODSACUDØ2) to the requirements file used in the system (ODSACUDØ1). Additionally, it initializes the cumulative data base input files (ODCUMUDØ1 and ODSAPUD18).

PFCAA.START Runstream:

```
@ASG,A ODSACUD02.             . START OF ODSAS PROCESSING
@ASG,A ODSACUD01.
@COPY ODSACUD02.,ODSACUD01.         . COPY INITIALIZAION FILE (REQUIREMENTS)
@FREE ODSACUD02.
@ASG,A ODINPUD01.
@ASG,A ODINPUD02.
@COPY ODINPUD02.,ODINPUD01.               . COPY INITIALIZATION FILE (INPUT)
@FREE ODINPUD02.
      @ASG,A ODCUMUD01.
      @ED,I ODCUMUD01.          . CLEAR CUMSEG DATABASE
      @ASG,A ODSAPUD18.
      @ED,I ODSAPUD18.          . CLEAR CUMSEG DATABASE
      @FREE ODCUMUD01.
      @FREE ODSAPUD18.
@ASG,A PFPRINT.
@PACK PFPRINT.
```

3. Processing Phase Runstreams

a. General Characteristics. - Each processing phase catalogued runstream has the following general characteristics:

(1) A unique run-ID number, ending in the grade segment number and an alphabetic character identifying segment (e.g., ODØØ6A is the run-ID for COLs (0-6) segment 1).

(2) All printing is output to a BRKPT file (ODØØ1UD9Ø or ODØØ1UD95). At the completion of each processing phase, the BRKPT file is copied to an element in the PFPRINT file for retention.

(3) The commands to be executed upon a normal termination in FMPS are placed in the element named "RUN" in TPF$. An abnormal termination in FMPS causes TPF$ to be @FREE'd, and thus the next commands to be processed are unavailable and processing ceases.

(4) A standard sequence of catalogued runstreams

(a) PFCAA.MATRIX-MODUL (matrix generator)*

(b) PFCAA.FREE (frees nonessential files before starting FMPS solution)

(c) PFCAA.FMPS-MODULE (FMPS solution for COL-MAJ segments)

---

*Element names are limited to 12 characters in length. MATRIX-MODUL without an "E" at the end of "MODUL" is correct.

or:

PFCAA.FMPS-MODULE/CPT-LT (FMPS solution for CPT and LT segments)

(d) PFCAA.DB-MODULE (data base creation)

(e) PFCAA.IMPLEMENT (load MIRADS data bases)

(5) An optional runstream, PFCAA.SAVECOPY/TOTAPE, may be @ADD'd to save copies on tape of the files that are essential to start a processing phase again after subsequent processing phases have been completed. In order to re-start a processing phase, the PFCAA.SAVECOPY/TODISC runstream (described at Paragraph 4a(10) below) must be @ADD'd to copy the appropriate tape to disc.

(6) A segment identification number is placed in an element named "SEGMENT" in TPF$. This number is used in the data base creation activity.

(7) Messages are sent to run-ID "ODØØ1_" on entering, and on completing, the FMPS module. This is implemented through the MAIL command in the ED processor.

b. <u>Unsegmented Grade Processing</u>. - The following runstreams would be employed for processing the five grade segments (the field grades in an unsegmented mode, followed by the CPT and LT segments):

(1) <u>PFCAA.RUN-COLXX</u>. - Specifies all commands to process an unsegmented COL run.

```
@RUN,/TP OD0U6X,0901112,CAA,500,1000
@TYPE 2,BA
@PRT,I
@ASG,A OD001UD90.
@BRKPT PRINT$/OD001UD90
@HDG,P COLONEL UNSEGMENTED  UNCL                          UNCLASSIFIED
@ELT,IO RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ASG,A ODINPUD001.
@ASG,A PFCAA.
@ED,I SEGMENT
61
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE OD001UD90.
@ASG,A OD001UD90.
@ASG,A PFPRINT.
@COPY,I OD001UD90.,PFPRINT.COLAX
@SYM,SU OD001UD90.,,PR
@FIN
```

(2) PFCAA.RUN-LTCYY. - Specifies all commands to process an unsegmented LTC run.

```
@RUN,/TP 00005Y,090112,CAA,3000,2000
@TYPE 2,BA
@ASG,A 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P     LTC UNSEGMENTED   ODSAS  CALL CPT TOFFLER     325-8152
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
51
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.LTCYY
@SYM,SU 00001UD95.,,PR
@FIN
```

(3) PFCAA.RUN-MAJZZ. - Specifies all commands to process an unsegmented MAJ run.

```
@RUN,/TP 00004Z,090112,CAA,2000,1000
@TYPE 2,BA
@ASG,A 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P     MAJOR UNSEGMENTED   ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
41
@PRT,I
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.MAJZZ
@SYM,SU 00001UD95.,,PR
@FIN
```

(4) PFCAA.RUN-CPT. - Specifies all commands to process a CPT segment.

```
@RUN,/TP 00003G,090112,CAA,1000,1000
@TYPE 2,BA
@ASG,A 00001UD90.
@BRKPT PRINT$/00001UD90
@HDG,P CAPTAIN SEGMENT   ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
```

PFCAA.RUN-CPT Runstream (continued)

```
1I
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE/CPT-LT
@ADD,L RUN
@BRKPT PRINT$
@FREE 0D0D1UD90.
@ASG,A 0D0D1UD90.
@ASG,A PFPRINT.
@COPY,I 0D0D1UD90.,PFPRINT.CPT
@SYM,SU 0D0D1UD90.,,PR
@FIN
```

(5) PFCAA.RUN-LT. - Specifies all commands to process a LT segment.

```
@RUN,/TP 0D002H,090112,CAA,500,1000
@TYPE 2,BA
@ASG,A 0D0D1UD90.
@BRKPT PRINT$/0D0D1UD90
@HDG,P LT SEGMENT          UNCL                          UNCLASSIFIED
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@ADD,L PFCAA.MESSAGE
@END
@ED,I SEGMENT
2I
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE/CPT-LT
@ADD,L RUN
@BRKPT PRINT$
@FREE 0D0D1UD90.
@ASG,A 0D0D1UD90.
@ASG,A PFPRINT.
@COPY,I 0D0D1UD90.,PFPRINT.LT
@SYM,SU 0D0D1UD90.,,PR
@FIN
```

c. Segmentation-Within-Grade, Colonel Segment. - The following runstreams would be substituted for PFCAA.RUN-COLXX, described in Paragraph b(1) above, if the segmentation-within-grade option was selected for the COL processing:

(1) PFCAA.RUN-COL1. - Specifies all commands to process the first COL segment.

```
@RUN,/TPH 0D006A,090112,CAA,500,1000
@TYPE 2,BA
@DELETE,C 0D0D1UD95.
@ASG,UP 0D0D1UD95.
@BRKPT PRINT$/0D0D1UD95
@HDG,P COLONEL SEGMENT 1   ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
6I
```

PFCAA.RUN-COL1 Runstream (continued)

```
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.COL1
@SYM,SU 00001UD95.,,PR
@FIN
```

(2) PFCAA.RUN-COL2. - Specifies all commands to process the second COL segment.

```
@RUN,/TPR 00006B,090112,CAA,2000,1000
@TYPE 2,BA
@DELETE,C 00001UD95.
@ASG,UP 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P COLONEL SEGMENT 2     ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
62
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.COL2
@SYM,SU 00001UD95.,,PR
@FIN
```

d. Segmentation-Within-Grade, Lieutenant Colonel Segment. - The following runstreams would be substituted for PFCAA.RUN-LTCYY, described in Paragraph b(2) above, if the segmentation-within-grade option was selected for the LTC processing:

(1) PFCAA.RUN-LTC1. - Specifies all commands to process the first LTC segment.

```
@RUN,/TPR 00005A,090112,CAA,500,1000
@TYPE 2,BA
@DELETE,C 00001UD95.
@ASG,UP 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P LTC      SEGMENT 1     ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
```

PFCAA.RUN-LTC1 Runstream (continued)

```
51
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.LTC1
@SYM,SU 00001UD95.,,PR
@FIN
```

(2) PFCAA.RUN-LTC2. - Specifies all commands to process the second LTC segment.

```
@RUN,/TPRS 00005B,090112,CAA,900,2000
@TYPE 2,BA
@DELETE,C 00001UD90.
@ASG,UP 00001UD90.
@BRKPT PRINT$/00001UD90
@HDG,P LTC SEGMENT 2   ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
52
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD90.
@ASG,A 00001UD90.
@ASG,A PFPRINT.
@COPY,I 00001UD90.,PFPRINT.LTC2
@SYM,SU 00001UD90.,,PR
@FIN
```

e. Segmentation-Within-Grade, Major Segment. The following runstreams would be substituted for PFCAA.RUN-MAJZZ, described in Paragraph b(3) above, if the segmentation-within-grade option was selected for the MAJ processing:

(1) PFCAA.RUN-MAJ1. - Specifies all commands to process the first MAJ segment.

```
@RUN,/TPRS 00004A,090112,CAA,500,1000
@TYPE 2,BA
@ASG,A 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P MAJOR SEGMENT 1  ODSAS
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
```

PFCAA.RUN-MAJ1 Runstream (continued)

```
41
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.MAJ1
@SYM,SU 00001UD95.,,PR
@FIN
```

(2) PFCAA.RUN-MAJ2. - Specifies all commands to process the second MAJ segment.

```
@RUN,/TPR 0D0048,U90112,CAA,900,2000
@TYPE 2,BA
@ASG,UP 00001UD95.
@BRKPT PRINT$/00001UD95
@HDG,P MAJOR SEGMENT 2  ODSAS.
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ED,I SEGMENT
42
@ADD,L PFCAA.MATRIX-MODUL
@ADD,L PFCAA.FREE
@ADD,L PFCAA.FMPS-MODULE
@ADD,L RUN
@BRKPT PRINT$
@FREE 00001UD95.
@ASG,A 00001UD95.
@ASG,A PFPRINT.
@COPY,I 00001UD95.,PFPRINT.MAJ2
@SYM,SU 00001UD95.,,PR
@FIN
```

f. Standard Runstreams Within the Unsegmented and Segmented Grade Processing Runstreams. - *The following runstreams are the* standard runstreams that are referred to within each of the processing phase runstreams listed above.

(1) PFCAA.LINK-MODULE. - Assigns all files, and executes the linkage program to update the specialty requirements file (ODSACUDØ1), creates an input parameter file (ODINPUDØ1), for the next system segment, and produces the unfilled requirements report.

PFCAA.LINK-MODULE Runstream:

```
@ASG,A ODDOIUD90.
@BRKPT PRINT$/ODDOIUD90
@HDG,P      ODSAS LINK-MODULE
@ASG,A PFODSAP.                              . BEGIN LINK-MODULE
@ASG,A ODSAPUD03.
@ASG,A ODSAPUD18.
@COPY ODSAPUD03.,ODSAPUD18.           . UPDATES CUMSEG DATABASE
@FREE ODSAPUD03.
@FREE ODSAPUD18.
@ASG,A ODSOLUD1.
@USE 10.,ODSOLUD1.
@ASG,A ODPOPUD01.
@USE 12.,ODPOPUD01.
@ASG,A ODINPUD01.
@FREE,A 11.
@ASG,T 11.
@ASG,A ODSACUD01.
@USE 14.,ODSACUD01.
@XQT,D PFODSAP.LINKABS
@ADD,P ODINPUD01.
@PMD,E
@COPY 11,ODINPUD01.
@DATA,L ODINPUD01.
@END
@BRKPT PRINT$
@FREE ODDOIUD90.
@SYM,SU ODDOIUD90.,,PR
```

(2) PFCAA.MATRIX-MODUL. - Assigns all files for, and executes the matrix generator program (which creates the equation file, (ODEQAUDØ1) for input to the FMPS programs).

```
@ASG,A ODSAPUD08.                       . BEGIN MATRIX-MODUL
@ASG,A ODSAPUD04.
@ASG,A PFODSAP.
@USE 8,ODSAPUD08.
@USE 4,ODSAPUD04.
@ASG,A  ODEQAUD01.
@ASG,A ODINPUD01.
@DATA,L ODINPUD01.
@END
@ASG,A ODSACUD01.
@USE 11.,ODSACUD01.
@ASG,A ODRATUD01.
@USE 10,ODRATUD01.
@USE 9,ODEQAUD01.
@XQT        PFODSAP.ABS       . MATRIX GENERATOR
@ADD,P ODINPUD01.
@PMD,E
```

(3) PFCAA.FREE. - Releases all nonessential files, so that FMPS programs can execute with a minimum of system overhead (if too many files are assigned to a runstream, random errors can abort an FMPS run).

```
@FREE 4.                  . BEGIN FREE MODULE
@FREE 8
@FREE 9
@FREE 10
@FREE 11
@FREE 12
@FREE 13
@FREE 14
```

(4) PFCAA.FMPS-MODULE. - Assigns all files, sends a message that the module has been entered, executes the FMPS program to solve the LP problem for the COL, LTC, and MAJ segments, and sends a message that the module has completed processing.

```
@ADD,P PFCAA.MESSAGE-ONE          . BEGIN FMPS-MODULE
@ASG,A ODEQAUD01.
@ASG,A ODSOLUD1.
@ASG,A PFFMPS.
@ASG,A ODDBSUD1.
@ASG,A  ODSAVUD01.
@ASG,A  ODRECUD01.
@USE 13,ODDBSUD1.
@USE 10.,ODSOLUD1.
@USE 11.,ODEQAUD01.
@USE 12.,ODSAVUD01.
@USE FMPS.,PFFMPS.
@USE SPRECVR.,ODRECUD01.
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P INVERSE.,F2/640/TRK/3000
@ASG,A INVERSE.
@ASG,T DUM2.,F2/10/POS/1000
@FREE DUM2.
@CAT,P MATRIX.,F2/640/TRK/3000
@ASG,A MATRIX.
@ASG,T DUM3.,F2/10/POS/1000
@FREE DUM3.
@CAT,P UTIL1.,F2/640/TRK/3000
@ASG,A UTIL1.
@ASG,T  DUM4.,F2/10/POS/1000
@FREE DUM4.
@CAT,P UTIL2.,F2/640/TRK/3000
@ASG,A UTIL2.
@XQT FMPS.LARGEFMPSABS
@ADD,P PFCAA.FMPS-CONTROL
@ADD ODEQAUD01.
 .  ROW SELECTION MASKS FOR DATA BASE
NAME            MASKDATM
MASKS
          N...CREW
          N...TREQ
          ....UBSG
          TOTAUTH.
ENDATA
 .
@ADD,L PFCAA.MESSAGE-TWO
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@FREE ODEQAUD01.
@FREE INVERSE.
@FREE UTIL1.
@FREE MATRIX.
@FREE UTIL2.
@FREE ODSOLUD1.
@FREE PFFMPS.
@FREE ODDBSUD1.
@FREE  ODSAVUD01.
@FREE  ODRECUD01.
```

(5) PFCAA.FMPS-MODULE/CPT-LT. - Assigns all files, sends a message that the module has been entered, executes the FMPS programs to solve the LP problem for the CPT and LT segments, and sends a message that the module has completed processing. The difference between this runstream and PFCAA.FMPS-MODULE above, is that different FMPS control statements (in PFCAA.FMPS-CONTROL/CPT-LT, Subparagraph (9) below) are employed in the CPT and LT segments.

```
@ADD,L PFCAA.MESSAGE-ONE
@ASG,A ODEQAUD01.                         . BEGIN FMPS-MODULE FOR CPT-LT
@ASG,A ODSOLUD1.
@ASG,A PFFMPS.
@ASG,A ODDBSUD1.
@ASG,A  ODSAVUD01.
@ASG,A  ODRECUD01.
@USE 13,ODDBSUD1.
@USE 10.,ODSOLUD1.
@USE 11.,ODEQAUD01.
@USE 12.,ODSAVUD01.
@USE FMPS.,PFFMPS.
@USE SPRECVR.,ODRECUD01.
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P INVERSE.,F2/640/TRK/3000
@ASG,A INVERSE.
@ASG,T DUM2.,F2/10/POS/1000
@FREE DUM2.
@CAT,P MATRIX.,F2/640/TRK/3000
@ASG,A MATRIX.
@ASG,T DUM3.,F2/10/POS/1000
@FREE DUM3.
@CAT,P UTIL1.,F2/640/TRK/3000
@ASG,A UTIL1.
@ASG,T  DUM4.,F2/10/POS/1000
@FREE DUM4.
@CAT,P UTIL2.,F2/640/TRK/3000
@ASG,A UTIL2.
@XQT FMPS.SMALLFMPSABS
@ADD,P PFCAA.FMPS-CONTROL/CPT-LT
@ADD ODEQAUD01.
ENDATA
   . ROW SELECTION MASKS FOR DATA BASE
NAME            MASKBASE
MASKS
          N...CHEQ
          N...THEQ
          RES.....
          TOTAUTH.
ENDATA
   .
@ADD,L PFCAA.MESSAGE-TWO
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@FREE ODEQAUD01.
@FREE INVERSE.
@FREE UTIL1.
@FREE MATRIX.
@FREE UTIL2.
@FREE ODSOLUD1.
@FREE PFFMPS.
@FREE ODDBSUD1.
@FREE  ODSAVUD01.
@FREE  ODRECUD01.
```

(6) PFCAA.DB-MODULE. - Assigns all files and executes the programs to create the input data base files for the current segment data base and the cumulative data base.

```
@ASG,A  ODDBSUD1.                              . BEGIN DATA BASE MODULE
@ASG,A  ODSAPUD03
@ASG,A  ODSAPUD04
@ASG,A  ODSAPUD07
@ASG,A  ODSAPUD18
@DELETE,C 10TEMP.
@PRT,I
@PRT,F ODSOLUD1.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P 10TEMP.,F2/640/TRK/3000
@ASG,A 10TEMP.
@USE 10.,10TEMP.
@USE 11,ODDBSUD1
@USE  3,ODSAPUD03
@USE  4,ODSAPUD04
@USE  7,ODSAPUD07
@USE 18,ODSAPUD18
@XQT PFODSAP.DATABASE
@ADD,LP SEGMENT
@PMD,E
@FREE 3
@FREE 4
@FREE 7
@FREE 10
@FREE 11
@FREE 18
@ASG,A ODSAPUD03.
@ASG,T 20.
@ASG,T 21.
@ASG,T 22.,F60///1000
@ASG,A ODINPUD01.
@USE 16.,ODINPUD01.
                     . LOCATE BEGINNING OF COLUMN RECORDS
                . PUT ROW RECORDS IN FILE 20.
                     . GO TO END OF FILE
               . PUT COLUMN RECORDS IN FILE 21.
                     . LEAVE ODSAPUD03 UNCHANGED
@ED,U ODSAPUD03.
F CW
SPLIT 20.
LAST
SPLIT! 21.
OMIT
@XQT PFODSAP.DB-CORRECT
@FREE 16.
@FREE 20.
@FREE 21.
@TEST TE/6/TJ            . SKIP NEXT STMT IF GRADE = 6
@COPY 22.,ODSAPUD03.
@FREE 22.
```

(7) PFCAA.IMPLEMENT. - Assigns and creates all files needed by the MIRADS system to load the current segment and cumulative data bases and prepare them for on-line inquiry.

PFCAA.IMPLEMENT Runstream:

```
@HDG,P     MIRADS IMPLEMENTATION  FOR ODSAS
@ASG,A PFMIRADS.          . BEGIN IMPLEMENT MODULE
@DELETE,C DICCUMSEG.
@DELETE,C SAVCUMSEG.
@DELETE,C MASCURSEG.
@DELETE,C DRLCUMSEG.
@DELETE,C INDCUMSEG.
@DELETE,C DICCUMSEG.
@DELETE,C SAVCUMSEG.
@DELETE,C MASCUMSEG.
@DELETE,C DRLCUMSEG.
@DELETE,C INDCUMSEG.
@XQT,C PFMIRADS,ASGFILS
CUMSEG
@ASG,A ODSAPUD07.
@USE 9.,ODSAPUD07.
@XQT PFODSAP.DBGEN
@XQT,SU PFMIRADS.DICGEN
@ADD,P PFODSAP.CUR-DICT
@XQT PFMIRADS.DRLGEN
@XQT PFMIRADS.INDGEN
@XQT PFMIRADS.SAVGEN
@ADD,P PFCAA.STD-QUERY/CUMSEG
@XQT,C PFMIRADS.ASGFILS
CUMSEG
@ASG,A ODSAPUD03.
@USE 9.,ODSAPUD03.
@XQT PFODSAP.DBGEN
@XQT,SU PFMIRADS.DICGEN
@ADD,P PFODSAP.CUM-DICT
@XQT PFMIRADS.DRLGEN
@XQT PFMIRADS.INDGEN
@XQT PFMIRADS.SAVGEN
@ADD,P PFCAA.STD-QUERY/CUMSEG
```

(8) PFCAA.FMPS-CONTROL. - Specifies the type and sequence of FMPS control statements used to direct the FMPS programs in arriving at an LP solution for the COL, LTC, and MAJ segments.

```
C       DEFINE PAGE TITLE
1       TITLE  ODSAS FMPS COL - MAJ
C       LP PROBLEM
2       CALL ENTER(LP,DUBLFMPS)
C       LOAD THE INPUT DECK FOR LP MATRIX
        CALL INPUT
C           MATRIX,INVERSE,UTIL1 AND UTIL2 FILES ATTACHED VIA @ASG
C       STANDARD TOLERANCE SETTINGS EXCEPT THE FOLLOWING
C           ......TOLERANCE ADJUSTMENTS.....
         IBARSPRN=-20
         IBDJSPRN=-20
C        ................       ................
C
C        INITIALIZE MAJOR, MINOR, AND I-O ERROR INTERRUPTS
3       ASSIGN 1000 TO KMAJER
4       ASSIGN 1010 TO KMINER
        ASSIGN 1020 TO KIOER
C       INITIALIZE SPRINT INTERRUPTS FOR  KNFS AND KUBS
        ASSIGN 1030 TO KNFS
         ASSIGN 1040 TO KUBS
C       SET INVERSION FREQ TO SPRINT VALUE
C           ..IF ROWCNT GT 1500, SET IFREQI=100...
C           ..IF ROWCNT GT 3000, SET IFREQI=150..
        IFREQI=150
C       SET TYPE OF PROBLEM AS MAX(-1), MIN(+1)
```

PFCAA.FMPS-CONTROL Runstream (continued)

```
      FOBJWT= -1.0
C       ..USER DEFINED INTERRUPT VARIABLE TO ENABLE CALL TO SPRINT
C       ...AFTER KFREQA INTERRUPT .....
        ASSIGN 20 TO KN01
      ASSIGN 1080 TO KFREQA
C     TO SAVE THE BASIS AFTER FIRST ITERATION, IN CASE NEED TO RESTART
      IFREQA=1
C      .......     ......     ......     .....    .....    ....
C     SETS DRIVE FOR OPTIMALITY .GT. DRIVE FOR FEASIBILITY
      FCMPDJ= 0.7
C      .......     ......     ......     .....    .....    ....
C     ASSIGN RESTART FILE AND ATTACH TO THIS RUN
 10     CALL ATTACH(RESTART,12)
C       ..ASSIGN SOLUTION FILE FOR LINKAGE TO NEXT SEGMENT..
      CALL ATTACH('ODSOLUDI',13,FORTRAN,NEW)
C        ..ASSIGN SOLUTION FILE FOR DATABASE INPUT
      CALL ATTACH('ODDBSUDI',13,FORTRAN,NEW)
C     ...LOADLISTS FOR SOLUTION FILE
      ADATA = 'MASKDATA'
      CALL LOADLIST(LISTR)
       ADATA = 'MASKDATH'
       CALL LOADLIST(LISTC)
C           CALL OUTPUT(BYROWS)        TO DISPLAY ALL EQUATIONS
C       DISPLAY INITIAL SETTINGS OF CR VARIABLES ..........
      CALL CONDITION
C      SOLVE THE PROBLEM ALLOWING SPRINT TO DEFINE BASIS
C
      CALL SPRINT(NOBASIS)
      GO TO KN01
 20    CALL SOLUTION

C           SAVE OPTIMAL SOLUTION
        CALL SAVE
      CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUDI',RCHAPTER,
     12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      ..... LOADLISTS FOR DATA BASE FILE.....
       ADATA='MASKDATH'
      CALL LOADLIST(LISTC,LIST)
C
C
C          ......FOLLOWING COMMENT STMTS NON-OP IN 6RIC....
C          LOAD MASKS FOR ROWS IN DATABASE FILE
C        CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C     CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
C    1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C         ............................................
      CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
C
      STOP
C      .......     ......     ......     .....    .....    ....
C     FOLLOWING ARE STMTS FOR THE INTERRUPTS
C     MAJER/MINER
 1000 CALL CONDITION
C       ...FREE NESTED RUNSTREAM UPON ABNORMAL TERMINATION
      CALL CSF('CFREE TPFS.' )
      STOP
 1010     CALL SOLUTION
C           SAVE OPTIMAL SOLUTION
        CALL SAVE
      CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUDI',RCHAPTER,
     12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      ..... LOADLISTS FOR DATA BASE FILE.....
       ADATA='MASKDATH'
      CALL LOADLIST(LISTC,LIST)
C
```

PFCAA.FMPS-CONTROL Runstream (continued)

```
C
C           ******FOLLOWING COMMENT STMTS NON-OP IN ARIC*****
C           LOAD MASKS FOR ROWS IN DATABASE FILE
C         CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C       CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
C     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C          ****************************************************
        CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
      1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
C
        STOP
C          **PROCEDURES FOR I/O ERRORS***
1020    GO TO 1000
C
C       INTERRUPTS FOR RNFS,RUBS
1030    CALL SOLUTION
        CALL OUTPUT(BYROWS,ROWS,LISTI)
C          ***FREE NESTED RUNSTREAM UPON ABNORMAL TERMINATION

        CALL CSF('@FREE TPFS.')
        STOP
1040    CALL OUTPUT(BYCOLS,COLS,LISTU)
          CALL SOLUTION
C          ***FREE NESTED RUNSTREAM UPON ABNORMAL TERMINATION
        CALL CSF('@FREE TPFS.' )
        STOP
1050    IINGHT=0
        ASSIGN 1070 TO KW01
         CALL CONDITION
          CALL SAVE
           CALL RESTORE
         EXIT
1070      ASSIGN 20 TO KW01
1075      CALL SPRINT
          GO TO KW01
1080    CALL SAVE
        CALL RESTORE
        ASSIGN 1050 TO KFREQA
C       NOW SAVE BASIS AFTER EVERY 1000 ITERATIONS
        IFREQA=1000
        CALL CONDITION
        RETURN
C       END OF CONTROL PROGRAM
 2000   END
```

(9) PFCAA.FMPS-CONTROL/CPT-LT. - Specifies the type and sequence of FMPS control statements used to direct the FMPS programs in arriving at an LP solution for the CPT and LT segments.

```
C      DEFINE PAGE TITLE
1      TITLE  ODSAS FMPS CPT - LT
C      LP PROBLEM
2      CALL ENTER(LP,DUBLFMPS)
C      LOAD THE INPUT DECK FOR LP MATRIX
       CALL INPUT
C      STANDARD ATTACH FOR MATRIX,INVERSE,UTIL1 AND UTIL2 .
C      STANDARD TOLERANCE SETTINGS EXCEPT THE FOLLOWING
C          ******TOLERANCE ADJUSTMENTS*****
        IBABSPRN=-20
        IBDJSPRN=-20
C       **************          ****************
C
C       INITIALIZE MAJOR AND MINOR ERROR INTERRUPTS
3      ASSIGN 1000 TO KMAJER
4      ASSIGN 1010 TO KMINER
       ASSIGN 1020 TO KIOER
C      INTIALIZE SPRINT INTERRUPTS KNFS,KUBS
       ASSIGN 1030 TO KNFS
        ASSIGN 1040 TO KUBS
C       SAVE RESULT EVERY ITIME=35 CPU MIN.
C      ASSIGN 1060 TO KTIME        DELETED JAN 76 JDT
C      ITIME=35
C      SET INVERSION FREQ TO SPRINT VALUE
C           **IF ROWCNT GT 1500, SET IFREQI=100***
C           **IF ROWCNT GT 3000, SET IFREQI=150**
       IFREQI=150
C      SET TYPE OF PROBLEM AS MAX(-1), MIN(+1)
       FOBJWT= -1.0
C      WEIGHTS INFEASIBILITY SWITCH AT HIGH VALUE FOR FIRST 10 ITERATIONS
C        *****IINGHT= -1      DELETED 23 OCT 75 JDT*****
C        **USER DEFINED INTERRUPT VARIABLE TO ENABLE CALL TO SPRINT
C        ***AFTER KFREQA INTERRUPT *****
        ASSIGN 20 TO KWO1
       ASSIGN 1080 TO KFREQA .
C      TO SAVE THE BASIS AFTER FIRST 100 ITERATIONS
       IFREQA=1
C       *******    ******     ******      *****    *****      ****
C      SETS DRIVE FOR OPTIMALITY .GT. DRIVE FOR FEASIBILITY
       FCMPOJ= 0.7
C      DEFINE INPUT DECK, PROBLEM NAME,OBJ ROW AND RHS
C       *******    ******     ******      *****    *****      ****
C      ASSIGN RESTART FILE AND ATTACH TO THIS RUN
10       CALL ATTACH(RESTART,12)
C        **ASSIGN SOLUTION FILE FOR LINKAGE TO NEXT SEGMENT**
       CALL ATTACH('ODSOLU01',10,FORTRAN,NEW)
       CALL ATTACH('ODDBSU01',13,FORTRAN,NEW)
C      CALL ATTACH('ODEQAU001',11,CARD,INONLY)
C 11    CALL INPUT(FILE,'ODEQAU001')
C      **** LOADLISTS FOR SOLUTION FILE
       ADATA = 'MASKDATA'
       CALL LOADLIST(LISTR)
        ADATA = 'MASKDATH'
        CALL LOADLIST(LISTC)
C       MAKE INITIAL SAVE FILE

C       SOLVE THE PROBLEM ALLOWING SPRINT TO DEFINE BASIS
C
C        DISPLAY INITIAL SETTINGS OF CR VARIABLES  **********.
       CALL CONDITION
       CALL SPRINT(NOBASIS)
       GO TO KWO1
20      CALL SOLUTION
```

PFCAA.FMPS-CONTROL/CPT-LT Runstream (continued)

```
C            SAVE OPTIMAL SOLUTION
         CALL SAVE
        CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUD1',RCHAPTER,
       12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C        ***** LOADLISTS FOR DATA BASE FILES *****
         ADATA='MASKDATM'
        CALL LOADLIST(LISTC,LIST)
C
C
C             ******FOLLOWING COMMENT STMTS NON-OP IN 6RIC****
C             LOAD MASKS FOR ROWS IN DATABASE FILE
C           CALL LOADLIST(LISTR,LIST)
C              CALL CONDITION
C       CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
C      1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C            ************************************************
        CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
       1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
C
C
        STOP
C        *******    ******    ******    *****   *****    ****
C       FOLLOWING ARE STMTS FOR THE INTERRUPTS
C       MAJER/MINER
 1000   CALL CONDITION
        CALL CSF('@FREE TPFS.' )
        STOP
1010        CALL SOLUTION
C            SAVE OPTIMAL SOLUTION
         CALL SAVE
        CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUD1',RCHAPTER,
       12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C        ***** LOADLISTS FOR DATA BASE FILES *****
         ADATA='MASKDATM'
        CALL LOADLIST(LISTC,LIST)
C
C
C             ******FOLLOWING COMMENT STMTS NON-OP IN 6RIC****
C             LOAD MASKS FOR ROWS IN DATABASE FILE
C           CALL LOADLIST(LISTR,LIST)
C              CALL CONDITION
C       CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
C      1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C            ************************************************
        CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
       1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
C
        STOP
C          **PROCEDURES FOR I/O ERRORS***
1020    GO TO 1000
C
C       INTERRUPTS FOR KNFS,KUBS
1030    CALL SOLUTION
        CALL OUTPUT(BYROWS,ROWS,LISTI)
        CALL CSF('@FREE TPFS.')
        STOP
1045    CALL SOLUTION
        CALL CSF('@FREE TPFS.' )
        STOP
1040    CALL OUTPUT(BYCOLS,COLS,LISTU)
        GO TO 1045
1050    IIWGHT=0
        ASSIGN 1070 TO KWOI
         CALL CONDITION
```

PFCAA.FMPS-CONTROL/CPT-LT Runstream (continued)

```
          CALL SAVE
           CALL RESTORE
          EXIT
C1060      CALL CONDITION
C         CALL ERRORS
C         CALL SAVE
C         GO TO 1075
1070       ASSIGN 20 TO KW01
1075       CALL SPRINT
           GO TO KW01
1080      CALL SAVE
C          CALL CSF('@BRKPT PRINTS' )
          CALL RESTORE
          ASSIGN 1050 TO KFREQA
C         NOW SAVE BASIS AFTER EVERY 1000 ITERATIONS
          IFREQA=1000
          CALL CONDITION
          RETURN
C         END OF CONTROL PROGRAM
 2000 END
```

(10) PFCAA.SEP-MODULE/4, PFCAA.SEP-MODULE/3, and PFCAA.SEP-MODULE/2. - Separates the records in the cumulative data base input file (ODSAPUD18) into two groups. Group one is those records for grades that cannot be further updated (e.g., COL records can be updated only by the COL and LTC solutions, thus, at the beginning of the MAJ segment, COL records are separated from LTC records). Group two is those records that can be updated by subsequent solutions. Group one records are placed in file ODCUMUDØ1, and group two records remain in file ODSAPUD18. The version name of the SEP-MODULE element (e.g., in SEP.MODULE/4, the number 4 is the version name) indicates the grade segment when the runstream is to be executed. The runstreams for the versions differ in the segment identifier loaded in the element named SEGMENT in TPF$.

PFCAA.SEP-MODULE/4 Runstream:

```
@ASG,A ODDQ1UD95.
@BRKPT PRINTS/ODDQ1UD95
@HDG,P     ODSAS SEP-MODULE FOR GRADE 4
@ED,I SEGMENT
41
@ASG,A PFODSAP                      . BEGIN SEP-MODULE
@ASG,A ODCUMUD01
@DELETE,C 16.
@DELETE,C 17.
@ASG,UP 16
@ASG,UP 17.
@ASG,A ODSAPUD18.
@USE 18,ODSAPUD18.
@XQT PFODSAP.SEPARATE
@ADD,P SEGMENT
@PMD,E
@ED,U ODCUMUD01.        . APPEND NON-CURRENT CUMSEG RECDS TO EOF
ADD 17.
EXIT
```

PFCAA.SEP-MODULE/4 Runstream (continued)

```
@COPY 16.,18.      . COPY CURRENT CUMSEG RECORDS TO FILE 18.
@DELETE,C 16.
@DATA,L 17.
@END
@DELETE,C 17.
@FREE 16
@FREE 17
@FREE 18
@FREE PFODSAP
@BRKPT PRINT$
@FREE ODOO1UD95.
@SYM,SU ODOO1UD95.,,PR
```

PFCAA.SEP-MODULE/3 Runstream:

```
@ASG,A ODOO1UD95.
@BRKPT PRINT$/ODOO1UD95
@HDG,P     ODSAS SEP-MODULE FOR GRADE 3
@ED,I SEGMENT
31
@ASG,A PFODSAP                   . BEGIN SEP-MODULE
@ASG,A ODCUMUDO1
@DELETE,C 16.
@DELETE,C 17.
@ASG,UP 16
@ASG,UP 17.
@ASG,A ODSAPUD18.
@USE 18,ODSAPUD18.
@XQT PFODSAP.SEPARATE
@ADD,P SEGMENT
@PMD,E
@ED,U ODCUMUDO1.      . APPEND NON-CURRENT CUMSEG RECDS TO EOF
ADD 17.
EXIT
@COPY 16.,18.      . COPY CURRENT CUMSEG RECORDS TO FILE 18.
@DELETE,C 16.
@DATA,L 17.
@END
@DELETE,C 17.
@FREE 16
@FREE 17
@FREE 18
@FREE PFODSAP
@BRKPT PRINT$
@FREE ODOO1UD95.
@SYM,SU ODOO1UD95.,,PR
```

PFCAA.SEP-MODULE/2 Runstream:

```
@ASG,A ODOO1UD95.
@BRKPT PRINT$/ODOO1UD95
@HDG,P     ODSAS SEP-MODULE FOR GRADE 2
@ED,I SEGMENT
21
@ASG,A PFODSAP                   . BEGIN SEP-MODULE
@ASG,A ODCUMUDO1
@DELETE,C 16.
@DELETE,C 17.
@ASG,UP 16
@ASG,UP 17.
@ASG,A ODSAPUD18.
@USE 18,ODSAPUD18.
```

PFCAA.SEP-MODULE/2 Runstream (continued)

```
@XQT PFODSAP.SEPARATE
@ADD,P SEGMENT
@PMD,E
@ED,U ODCUMUD01.        . APPEND NON-CURRENT CUMSEG RECDS TO EOF
ADD 17.
EXIT
@COPY 16.,18.     . COPY CURRENT CUMSEG RECORDS TO FILE 18.
@DELETE,C 16.
@DATA,L 17.
@END
@DELETE,C 17.
@FREE 16
@FREE 17
@FREE 18
@FREE PFODSAP
@BRKPT PRINT$
@FREE ODOO1UD95.
@SYM,SU ODOO1UD95.,,PR
```

(11) PFCAA.SEP-MODULE/1. - Functions of this runstream are similar to those in Subparagraph (10) above except: (1) this runstream will place the remaining records in ODSAPUD18 (i.e., CPT and LT records) in the ODCUMUDØ1 file, and (2) load the cumulative data base containing all grade records.

```
@ASG,A ODOO1UD95.
@BRKPT PRINT$/ODOO1UD95
@HDG,P    ODSAS SEP-MODULE TO PRODUCE FINAL CUMSEG DATA BASE
@ED,I SEGMENT
00
@ASG,A PFODSAP                    . BEGIN SEP-MODULE
@ASG,A ODCUMUD01
@DELETE,C 16.
@DELETE,C 17.
@ASG,UP 16
@ASG,UP 17.
@ASG,A ODSAPUD18.
@USE 18,ODSAPUD18.
@XQT PFODSAP.SEPARATE
@ADD,P SEGMENT
@PMD,E
@ED,U ODCUMUD01.        . APPEND NON-CURRENT CUMSEG RECDS TO EOF
ADD 17.
EXIT
@COPY 16.,18.     . COPY CURRENT CUMSEG RECORDS TO FILE 18.
@DELETE,C 16.
@DATA,L 17.
@END
@DELETE,C 17.
@FREE 16
@FREE 17
@FREE 18
@ADD,L PFCAA.IMPLEMENT/TOTAL
@FREE PFODSAP
@BRKPT PRINT$
@FREE ODOO1UD95.
@SYM,SU ODOO1UD95.,,PR
```

(12) PFCAA.SAVECOPY/TOTAPE. - Copies the current version of eleven (11) disc files to an output tape for retention, so that a processing phase can be re-started after subsequent processing phases have been run.

```
@HDG,P  SAVE ODSAS PROCESSING PHASE FILES ON TAPE
@ASG,T TAPE.,16N,OUTPUT        . OUTPUT TAPE
@ASG,A ODINPUD01.       . I-P PARAMETERS
@ASG,A ODSACUD01.       . REQUIREMENTS
@ASG,A ODDBSUD1.        . FMPS O-P TO DB
@ASG,A ODSOLUD1.        . FMPS O-P TO LINKAGE
@ASG,A ODSAPUD18.       . CUM DB FILE
@ASG,A ODSAPUD07.       . CUMSEG DB FILE
@ASG,A ODSAPUD04.       . MG O-P TO DB
@ASG,A ODCUMUD01.       . PERM CUM DB FILE
@ASG,A ODSAPUD03.       . TEMP CUM DB FILE
@ASG,A ODRECUD01.       . FMPS RECOVERY FILE
@ASG,A ODSAVUD01.       . FMPS SAVE BASIS FILE
@COPY,GM ODINPUD01.,TAPE.
@COPY,GM ODSACUD01.,TAPE.
@COPY,GM ODDBSUD1.,TAPE.
@COPY,GM ODSOLUD1.,TAPE.
@COPY,GM ODSAPUD18.,TAPE.
@COPY,GM ODSAPUD07.,TAPE.
@COPY,GM ODSAPUD04.,TAPE.
@COPY,GM ODCUMUD01.,TAPE.
@COPY,GM ODSAPUD03.,TAPE.
@COPY,GM ODRECUD01.,TAPE.
@COPY,GM ODSAVUD01.,TAPE.
```

(13) PFCAA.PROCURE. - This runstream assigns the files and executes the programs needed to interface with the MIRADS "HITFILE" of query-selected data base records and produces the ODSAS procurement report at the completion of the LT segment.

```
@FREE 9.
@HDG.P      ODSAS PROCUREMENT RUN
@USE 12.,MASCUMSEG.
@USE 11.,HITFILE.
@XQT PFODSAP.HITFILE/INTERFACE
@COPY 9.,OD0001UD90.
@ASG.A OD0001UD90.
@USE 15.,OD0001UD90.
@ASG.A ODPOPUD01.
@USE 12.,ODPOPUD01.
@DELETE.C 90ALTPRNT.
@ASG.UP 90ALTPRNT.        . ALTERNATE PRINT FILE
@USE 7.,90ALTPRNT.
@XQT PFODSAP.PROCUREMENT
@ASG.A PFPRINT.
@COPY.I OD0001UD90.,PFPRINT.COLXX
@SYM.SU OD0001UD90.,,PR
@FIN
```

g. Status Messages to the MAIL File. - The last three runstreams in the processing phase send messages to a "MAIL" file so that the user can inquire into the status of the run through use of the ED processor (see Chapter VI, Operations Guide, Paragraph 10):

(1) PFCAA.MESSAGE-ONE. - Sends a message to run-ID ODØØ1 that the FMPS-Module was entered in the runstream ODØØ1a, where "a" identifies the specific runstream, (e.g., ODØØ1X identifies RUN-COLXX), from which the message was sent.

```
@ED,I X
MAIL ODOO1
FMPS MODULE ENTERED
@EOF
```

(2) PFCAA.MESSAGE-TWO. - Similar to (1) above, except message is sent at completion of FMPS-MODULE.

```
@ED,I X
MAIL ODOO1
FMPS SEGMENT COMPLETED
@EOF
OMIT
```

(3) PFCAA.MESSAGE. - Similar to (1) above, except message is sent at completion of the IMPLEMENT runstream in LT segment, indicating that end of ODSAS processing has been reached.

```
@ED,I X
MAIL ODOO1
LT SEGMENT COMPLETED
@EOF
OMIT
```

4. Ancillary Runstreams

a. Analysis and/or Correction of Results at Completion of a Processing Phase. - The following catalogued runstreams can be executed at the completion of a processing phase to analyze or correct the results of the segment just completed:

(1) PFCAA.UPDATE. - Assigns all files and executes the program (PFODSAP.UPDATE) to change requirements and/or attrition and promotion rates or other input parameters, and updates the appropriate files to re-run the current segment.

```
@RUN,/TP 0D00IU,09011269999,CAA
@DELETE,C 0D00IU090.
@ASG,UP 0D00IU090.
@BRKPT PRINT$/0D00IU090
@HDG,P    ODSAS FILE UPDATE
@ASG,A ODSAC0001.
@ASG,A ODPOP0001.
@XQT PFODSAP.UPDATE
@ADD,P PFODSAP.UPDATE/CARDS
@BRKPT PRINT$
@FREE 0D00IU090.
@SYM,SU 0D00IU090.,,PR
@FIN
```

(2) PFCAA.MODIFY. - Specifies all commands to modify a current FMPS solution for COLs, LTCs and MAJs and solve again from an intermediate point in the solution process. The data elements to be modified are in the element named "MODIFY/CARDS" within the file named PFCAA (see line 42 in the runstream below). In order to modify a CPT or LT segment FMPS solution, the changes should be inserted in the element named "MODIFY/CARDS-CPT-LT" (see Paragraph (4) below), and the PFCAA.MODIFY runstream changed via the ED processor (change ".MODIFY/CARDS" to ".MODIFY/CARDS-CPT-LT").

```
@RUN,/TPR 0D00IP,090112,CAA,500,1000
@TYPE 2,8A
@ASG,A  0D00IU090.
@BRKPT PRINT$/0D00IU090
@HDG,P PFCAA/MODIFY    ODSAS    CPT TOFFLER 5-8250
@ED,I SEGMENT
51
@ELT,ID RUN
@ADD,L PFCAA.RE-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
@ASG,A ODEQA0001.
@ASG,A ODSOL0001.
@ASG,A PFFMPS.
@ASG,A ODDBS0001.
@USE 13,ODDBS0001.
@ASG,A ODREC0001.
@ASG,A ODSAV0001.
@ASG,A  ODREC0001.
@USE 10.,ODSOL0001.
@USE 11.,ODEQA0001.
```

PFCAA.MODIFY Runstream (continued)

```
@USE 12.,00SAV0001.
@USE FMPS.,PFFMPS.
@USE SPRECVR.,00RECU001.
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P INVERSE.,F2/640/TRK/3000
@ASG,A INVERSE.
@ASG,T DUM2.,F2/10/POS/1000
@FREE DUM2.
@CAT,P MATRIX.,F2/640/TRK/3000
@ASG,A MATRIX.
@ASG,T DUM3.,F2/10/POS/1000
@FREE DUM3.
@CAT,P UTIL1.,F2/640/TRK/3000
@ASG,A UTIL1.
@ASG,T  DUM4.,F2/10/POS/1000
@FREE DUM4.
@CAT,P UTIL2.,F2/640/TRK/3000
@ASG,A UTIL2.
@ADD,LP RUN
@ADD,P PFCAA.MESSAGE-ONE
@XQT FMPS.LARGEFMPSABS
@ADD,P PFCAA.FMPS-MODIFY
@ADD PFCAA.MODIFY/CARDS              . IF MODIFYING A CPT OR LT SEGMENT,
          . CHANGE ELEMENT NAME TO MODIFY/CARDS-CPT-LT, OTHERWISE
@ADD,P PFCAA.MESSAGE-TWO             . ELEMENT NAME SHOULD BE 'MODIFY/CARDS'
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@BRKPT PRINT$
@FREE 00001009O.
@SYM,SU 00001009O.,,PR
@FIN
```

(3) PFCAA.MODIFY/CARDS. - Specifies the data elements and values to be modified and the parameter values for the FMPS loadlist procedure in the COL, LTC, and MAJ segments. Below is an example of types of data elements and values that can be modified. The loadlist parameters are required to be in the type and form shown:

```
NAME           MODIFY
 UP  .BOUNDS.  X00021      108.000
ENDATA
NAME           MASKDATM
MASKS
          AA**
ENDATA
NAME           MASKDATM
MASKS
          N***CREQ
          N***TREQ
          ****OBSG
          TOTAUTH*
ENDATA
```

(4) PFCAA.MODIFY/CARDS-CPT-LT. - Specifies the data elements and values to be modified and the parameter values for the FMPS loadlist procedure in the CPT and LT segments. Below is an example of types of data elements and values that can be modified. The loadlist parameters are required to be in the type and form shown:

```
NAME            MODIFY
 LO .BOUNDS.    XN37        247.0
 UP .BOUNDS.    XN37        533.0
 LO .BOUNDS.    XN42        472.0
 UP .BOUNDS.    XN42        1504.0
 LO .BOUNDS.    XN43        52.0
 UP .BOUNDS.    XN43        134.0
 LO .BOUNDS.    XN44        157.0
 UP .BOUNDS.    XN44        362.0
 LO .BOUNDS.    XN71        117.0
 UP .BOUNDS.    XN71        603.0
ENDATA
NAME            MASKDATM
MASKS
        AA..
ENDATA
NAME            MASKDATM
MASKS
        N...CREQ
        N...TREQ
        RES......
        TOTAUTH.
ENDATA
```

(5) PFCAA.FMPS-MODIFY. - Specifies the type and sequence of FMPS control statements used to direct the FMPS programs to modify an existing solution and re-solve from an intermediate point in the solution process.

```
C       DEFINE PAGE TITLE
            TITLE     MODIFY OLD SOLUTION
C       LP PROBLEM
 2      CALL ENTER(LP,DOUBLFMPS)
C       STANDARD ATTACH FOR MATRIX,INVERSE,UTIL1 AND UTIL2
C       STANDARD TOLERANCE SETTINGS EXCEPT THE FOLLOWING
C           ......TOLERANCE ADJUSTMENTS.....
C         ...............          ................
C
C        INITIALIZE MAJOR AND MINOR ERROR INTERRUPTS
 3      ASSIGN 1000 TO KMAJER
 4      ASSIGN 1010 TO KMINER
        ASSIGN 1020 TO KIOER
C       INITIALIZE SPRINT INTERRUPTS KNFS,KUBS
        ASSIGN 1030 TO KNFS
         ASSIGN 1040 TO KUBS
C       ASSIGN RESTART FILE AND ATTACH TO THIS RUN
 10       CALL ATTACH(RESTART,12)
C         ..ASSIGN SOLUTION FILE FOR LINKAGE TO NEXT SEGMENT..
        CALL ATTACH('00SOLU01',10,FORTRAN,NEW)
        CALL ATTACH('000BSUD1',13,FORTRAN,NEW)
            CALL RESTORE
             IFREQ8=1500
              ASSIGN 1050 TO KFREQ8
          ADATA= 'MODIFY'
          CALL MODIFY
              CALL SPRINT
```

PFCAA.FMPS-MODIFY Runstream (continued)

```
C
      CALL CONDITION
20     CALL SOLUTION
C         SAVE OPTIMAL SOLUTION
          CALL SAVE
C     CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUDI',RCHAPTER,
C    12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      .....DUMMY LOADLIST TO INCLUDE ALL COLUMNS.....
       ADATA='MASKDATM'
      CALL LOADLIST(LISTC,LIST)
C          ......FOLLOWING COMMENT STMTS NON-OP IN 6R1C....
C          LOAD MASKS FOR ROWS IN DATABASE FILE
C        CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C     CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
C    1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C      ..............................................
      CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
      STOP
C      .......   ......    ......    .....   .....    ....
C     FOLLOWING ARE STMTS FOR THE INTERRUPTS
C     MAJER/MINER
 1000 CALL CONDITION
      CALL CSF('@FREE TPFS.' )
      STOP
1010     CALL SOLUTION
C         SAVE OPTIMAL SOLUTION
         CALL SAVE
      CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUDI',RCHAPTER,
     12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      .....DUMMY LOADLIST TO INCLUDE ALL COLUMNS.....
       ADATA='MASKDATM'
      CALL LOADLIST(LISTC,LIST)
C          ......FOLLOWING COMMENT STMTS NON-OP IN 6R1C....
C          LOAD MASKS FOR ROWS IN DATABASE FILE
C        CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C     CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
C    1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C      ..............................................
      CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUDI',
     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
      STOP
C       ..PROCEDURES FOR I/O ERRORS...
1020  GO TO 1000
C
C     INTERRUPTS FOR KNFS,KUBS
1030  CALL SOLUTION
      CALL OUTPUT(BYROWS,ROWS,LISTI)
      CALL CSF('@FREE TPFS.')
      STOP
1045  CALL SOLUTION
      CALL CSF('@FREE TPFS.' )
      STOP
1040  CALL OUTPUT(BYCOLS,COLS,LISTU)
      GO TO 1045
1050       GO TO 1060
1060   CALL CONDITION
       CALL SAVE
      CALL RESTORE
C     NOW SAVE BASIS AFTER EVERY 1000 ITERATIONS
      IFREQ0=1000
      RETURN
C     END OF CONTROL PROGRAM
 2000 END
```

(6) PFCAA.RECOVER. - Specifies all commands to recover an FMPS solution for COL, LTC, or MAJ segments that terminated prematurely because of hardware/software errors, or operator intervention.

```
@RUN,/TP 00001R,090112,CAA,2000,1000
@TYPE 2,8A
@ASG,A 000010090.
@BRKPT PRINT$/C00010090
@MSG,P        00SA9 RECOVERY OF FMPS RUN  CALL CPT TOFFLER  325-8152
@ASG,A PFFMPS.        . LEVEL 8R1C
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END
     .THE FOLLOWING GRADE SEGMENT INDICATOR MUST CORRESPOND
     .TO ORIGINAL GRADE-SEGMENT IDENTIFICATION
@ED.I SEGMENT

$1
@ASG,A 000BSU01.
@USE  13.,000BSU01.
@ASG,A 00RECU001.
@USE SPRECVR.,00RECU001.
@ASG,A 00SAVU001.
@USE 12.,00SAVU001.
@ASG,A 00SOLU01.
@USE 10.,00SOLU01.
@USE FMPS.,PFFMPS.
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@PRT,I
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P INVERSE.,F2/640/TRK/3000
@ASG,A INVERSE.
@ASG,T DUM2.,F2/10/POS/1000
@FREE DUM2.
@CAT,P MATRIX.,F2/640/TRK/3000
@ASG,A MATRIX.
@ASG,T DUM3.,F2/10/POS/1000
@FREE DUM3.
@CAT,P UTIL1.,F2/640/TRK/3000
@ASG,A UTIL1.
@ASG,T  DUM4.,F2/10/POS/1000
@FREE DUM4.
@CAT,P UTIL2.,F2/640/TRK/3000
@ASG,A UTIL2.
@ADD,L PFCAA.MESSAGE-ONE
@XQT   FMPS.LARGEFMPSABS
@ADD PFCAA.FMPS-RECOVER
  . COLUMN AND ROW SELECTION LISTS FOR DATABASE
NAME           MASKDATM
MASKS
         AA..
ENDATA
NAME           MASKDATM

MASKS
         N...CREW
         N...TREW
         N...UB56
         TOTAUTH.
ENDATA
  .
@PMD,ELP
@ADD,P PFCAA.MESSAGE-TWO
```

PFCAA.RECOVER Runstream (continued)

```
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@FREE ODE@AUD01.
@FREE INVERSE.
@FREE UTIL1.
@FREE MATRIX.
@FREE UTIL2.
@FREE ODSOLUD1.
@FREE PFFMPS.
@FREE ODDBSUD1.
@FREE  ODSAVUD01.
@FREE  ODRECUD01.
@ADD,L RUN
@BRKPT PRINT$
@FREE O0001UD90.
@SYM,SU O0001UD90...PR
@FREE O0001UD90.
@END
```

(7) PFCAA.RECOVER/CPT-LT. - Specifies all commands to recover an FMPS solution for a CPT or LT segment that terminated prematurely because of hardware/software errors, or operator intervention.

```
@RUN,/TPR O0001R,090112,CAA,2000,1000
@TYPE 2,BA
@ASG,A O0001UD95.
@BRKPT PRINT$/O0001UD95
@HDG,P      ODSAS RECOVERY OF FMPS RUN
@ASG,A PFFMPS.
@ELT,ID RUN
@ADD,L PFCAA.DB-MODULE
@ADD,L PFCAA.IMPLEMENT
@END                . THE FOLLOWING GRADE-SEGMENT INDICATOR MUST
@ED,I SEGMENT       . CORRESPOND TO ORIGINAL GRADE-SEGMENT IDENTIFICATION
31
@ASG,A ODDBSUD1.
@USE 13.,ODDBSUD1.
@ASG,A ODRECUD01.
@USE SPRECVR.,ODRECUD01.
@ASG,A ODSAVUD01.
@USE 12.,ODSAVUD01.
@ASG,A ODSOLUD1.
@USE 10.,ODSOLUD1.
@ADD,L PFCAA.MESSAGE-ONE
@USE FMPS.,PFFMPS.
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@ASG,T DUM1.,F2/10/POS/1000
@FREE DUM1.
@CAT,P INVERSE.,F2/640/TRK/3000
@ASG,A INVERSE.
@ASG,T DUM2.,F2/10/POS/1000
@FREE DUM2.
```

PFCAA.RECOVER/CPT-LT Runstream (continued)

```
@CAT,P MATRIX.,F2/640/TRK/3000
@ASG,A MATRIX.
@ASG,T DUM3.,F2/10/POS/1000
@FREE DUM3.
@CAT,P UTIL1.,F2/640/TRK/3000
@ASG,A UTIL1.
@ASG,T  DUM4.,F2/10/POS/1000
@FREE DUM4.
@CAT,P UTIL2.,F2/640/TRK/3000
@ASG,A UTIL2.
@XQT FMPS.MEDFMPSABS
@ADD PFCAA.FMPS-RECOVER
  . COLUMN AND ROW SELECTION LISTS FOR DATABASE
NAME            MASKDATM
MASKS
         AA..
ENDATA
NAME            MASKDATM
MASKS
         N...CREQ
         N...TREQ
         RES......

         TOTAUTH.
ENDATA
  .
@ADD,LP PFCAA.MESSAGE-TWO
@DELETE,C INVERSE.
@DELETE,C MATRIX.
@DELETE,C UTIL1.
@DELETE,C UTIL2.
@FREE ODEQAUD01.
@FREE INVERSE.
@FREE UTIL1.
@FREE MATRIX.
@FREE UTIL2.
@FREE ODSOLUD1.
@FREE PFFMPS.
@FREE ODDBSUD1.
@FREE  ODSAVUD01.
@FREE  ODRECUD01.
@ADD,L RUN
@BRKPT PRINTS
@FREE OD001UD95.
@SYM,SU OD001UD95.,,PR
@END
```

(8) PFCAA.FMPS-RECOVER. - Specifies the type and sequence of FMPS control statements used to direct the FMPS programs to recover the prematurely terminated segment processing of any segment (COL-LT) and begin the solution process at the point where the solution was last saved (on file ODRECUDØ1).

PFCAA.FMPS-RECOVER Runstream:

```
C     DEFINE PAGE TITLE
         TITLE    SPRINT RECOVERY
C     LP PROBLEM
 2    CALL ENTER(LP,DUBLFMPS)
C     STANDARD ATTACH FOR MATRIX,INVERSE,UTIL1 AND UTIL2
C     STANDARD TOLERANCE SETTINGS EXCEPT THE FOLLOWING
C        ******TOLERANCE ADJUSTMENTS*****
C      **************          ***************
C
C      INITIALIZE MAJOR AND MINOR ERROR INTERRUPTS
 3    ASSIGN 1000 TO KMAJER
 4    ASSIGN 1010 TO KMINER
      ASSIGN 1020 TO KIDER
C     INTIALIZE SPRINT INTERRUPTS KNFS,KUBS
      ASSIGN 1030 TO KNFS
       ASSIGN 1040 TO KUBS
C     ASSIGN RESTART FILE AND ATTACH TO THIS RUN
 10     CALL ATTACH(RESTART,12)
C       **ASSIGN SOLUTION FILE FOR LINKAGE TO NEXT SEGMENT**
      CALL ATTACH('ODSOLUD1',10,FORTRAN,NEW)
C         ASSIGN DATABASE INPUT FILE
      CALL ATTACH('ODDBSUD1',13,FORTRAN,NEW)
          CALL RESTORE
           IFREQA=1500
            ASSIGN 1050 TO KFREQA
C      CONTINUE SOLUTION FROM SPRINT RECOVERY FILE(SPRECVR)
              CALL SPRINT(NOBASIS,RECOVER)
C
      CALL CONDITION
 20    CALL SOLUTION
C         SAVE OPTIMAL SOLUTION
         CALL SAVE
      CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUD1',RCHAPTER,
     12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      *****DUMMMY LOADLIST TO INCLUDE ALL COLUMNS*****
        ADATA= 'MASKDATH'
      CALL LOADLIST(LISTC,LIST)
C            ******FOLLOWING COMMENT STMTS NON-OP IN GRIC****
C            LOAD MASKS FOR ROWS IN DATABASE FILE
C         CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C     CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
C    1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C          ****************************************************
      CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
      STOP
C      *******    ******     ******     *****    *****     ****
C     FOLLOWING ARE STMTS FOR THE INTERRUPTS
C     MAJER/MINER
 1000 CALL CONDITION
      CALL CSF('@FREE TPFS.' )
      STOP
 1010    CALL SOLUTION
C          SAVE OPTIMAL SOLUTION
         CALL SAVE
      CALL SOLUTION(ROWS,LISTR,COLS,LISTC,FILEONLY,'ODSOLUD1',RCHAPTER,
     12,3,4,6,7,CCHAPTER,2,3,4,6,7)
C      *****DUMMMY LOADLIST TO INCLUDE ALL COLUMNS*****
        ADATA= 'MASKDATH'
      CALL LOADLIST(LISTC,LIST)
C            ******FOLLOWING COMMENT STMTS NON-OP IN GRIC****
C            LOAD MASKS FOR ROWS IN DATABASE FILE
C         CALL LOADLIST(LISTR,LIST)
C            CALL CONDITION
C     CALL SOLUTION(ROWS,LISTR,COLS,EXCEPT,LISTC,FILEONLY,'ODDBSUD1',
C    1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
```

PFCAA.FMPS-RECOVER Runstream (continued)

```
C         ************************************************
      CALL SOLUTION(COLS,EXCEPT,LISTC,FILEONLY,'OUDBSUDI',
     1RCHAPTER,2,4,6,7,8,CCHAPTER,2,4,6,7,8)
C
      STOP
C        **PROCEDURES FOR I/O ERRORS***
1020  GO TO 1000
C
C     INTERRUPTS FOR RNFS,RUBS
1030  CALL SOLUTION
      CALL OUTPUT(BYROWS,ROWS,LISTI)
      CALL CSF('@FREE TPFS.')
      STOP
1040  CALL OUTPUT(BYCOLS,COLS,LISTU)
1045  CALL SOLUTION
      CALL CSF('@FREE TPFS.' )
      STOP
1050   CALL CONDITION
       CALL SAVE
      CALL RESTORE
C     NOW SAVE BASIS AFTER EVERY 1000 ITERATIONS
      IFREQA=1000
      RETURN
C     END OF CONTROL PROGRAM
2000 END
```

(9) PFCAA.IMPLEMENT/TOTAL. - Assigns and creates all files needed by the MIRADS system to load the cumulative data base containing all grade segments processed thus far, and prepare the data base for on-line inquiry.

```
@HDG,P    MIRADS IMPLEMENTATION  FOR ODSAS
@ASG,A PFMIRADS.           . BEGIN IMPLEMENT TOTAL MODULE
@DELETE,C DICCUMSEG.
@DELETE,C SAVCUMSEG.
@DELETE,C MASCUMSEG.
@DELETE,C DRLCUMSEG.
@DELETE,C INDCUMSEG.
@XQT,C PFMIRADS.ASGFILS
CUMSEG  .
@ASG,A ODCUMODUI.           . ALL CUMULATIVE DB RECORDS
@USE 9.,ODCUMODUI.
@XQT PFODSAP.DBGEN
@XQT,SU PFMIRADS.DICGEN
@ADD,P PFODSAP.CUM-DICT
@XQT PFMIRADS.DRLGEN
@XQT PFMIRADS.INDGEN
@XQT PFMIRADS.SAVGEN
@ADD,P PFCAA.STD-QUERY/CUMSEG
```

(10) PFCAA.SAVECOPY/TODISC. - This runstream enables restarting a processing phase after subsequent processing phases have been completed. The runstream copies a previously created tape containing copies of eleven ODSAS files to the appropriately named disc files. Thus, data are restored to the disc files in their original condition.

PFCAA.SAVECOPY/TODISC Runstream:

```
@HDG,P  COPY PAST PROCESSING PHASE FILES TO DISC
@ASG,T TAPE.,16N,••••••         . INPUT TAPE
@ASG,A ODINPUD01.        . I-P PARAMETERS
@ASG,A ODSACUD01.        . REQUIREMENTS
@ASG,A ODDBSUD1.         . FMPS O-P TO DB
@ASG,A ODSOLUD1.         . FMPS O-P TO LINKAGE
@ASG,A ODSAPUD18.        . CUM DB FILE
@ASG,A ODSAPUD07.        . CURSEG DB FILE
@ASG,A ODSAPUD04.        . MG O-P TO DB
@ASG,A ODCUMUD01.        . PERM CUM DB FILE
@ASG,A ODSAPUD03.        . TEMP CUM DB FILE
@ASG,A ODRECUD01.        . FMPS RECOVERY FILE
@ASG,A ODSAVUD01.        . FMPS SAVE BASIS FILE
@COPY,G TAPE.,ODINPUD01.
@COPY,G TAPE.,ODSACUD01.
@COPY,G TAPE.,ODDBSUD1.
@COPY,G TAPE.,ODSOLUD1.
@COPY,G TAPE.,ODSAPUD18.
@COPY,G TAPE.,ODSAPUD07.
@COPY,G TAPE.,ODSAPUD04.
@COPY,G TAPE.,ODCUMUD01.
@COPY,G TAPE.,ODSAPUD03.
@COPY,G TAPE.,ODRECUD01.
@COPY,G TAPE.,ODSAVUD01.
```

b. Saving the MIRADS Data Bases. - The following two runstreams support the MIRADS data bases, and provide the user with control over saving the data bases:

(1) PFCAA.ROLLOUT. - Copies the current segment (CURSEG) data base files and the cumulative (CUMSEG) data base file onto a tape file in order to save the data bases.

```
@ASG,T UFSTAP.,16N,06271        . MIRADS TAPE FILE
@XQT PFMIRADS.ROLLOUT
CUMSEG
@XQT PFMIRADS.ROLLOUT
CURSEG
@END
```

(2) PFCAA.ROLLIN. - Copies the files from the tape created in PFCAA.ROLLOUT, back to disc files so that the current segment (CURSEG) data base and the cumulative (CUMSEG) data base can be accessed via MIRADS.

```
@ASG,T UFSTAP.,16N,06271      .MIRADS TAPE FILE
@XQT PFMIRADS.ROLLIN
CUMSEG
@XQT PFMIRADS.ROLLIN
CURSEG
@END
```

c. Standard Query Sets. - The following two elements contain the standard query sets input to the runstream PFCAA.IMPLEMENT. As additional standard query sets are identified, they may be appended to the end of the element via the ED processor with a first card image of the following form:

SAV. "query set name"

The last card image in the element must be @END.

(1) PFCAA.STD-QUERY/CURSEG. - Contains all the standard MIRADS query sets used for the current segment (CURSEG) data base. These queries are loaded to the SAVCURSEG file by the MIRADS program, SAVGEN, in the PFCAA.IMPLEMENT runstream.

```
SAV.NODECAP
Q,ID = R AND PREFIX = N AND YEAR GE 0 AND FROM GT 0
   AND ACTIVITY EQ =UPLIMIT
S,YEAR,FROM
P,YEAR,FROM,ACTIVITY,UPLIMIT
SAV.NOTCAPTO
Q,ID = R AND PREFIX = N AND YEAR EQ 0 AND FROM GT 0
   AND ACTIVITY NE =UPLIMIT
C,ALL,3,$FILL% = ACTIVITY / UPLIMIT
S,YEAR,FROM
P,YEAR,FROM,ACTIVITY,UPLIMIT,$FILL%
SAV.CPTDESIGN
Q,ID = R AND PREFIX = V AND YEAR GE 0 AND ACTIVITY GT 0
S,YEAR,FROM,TO
P,YEAR,FROM,TO,ACTIVITY
SAV.SPEC-PAIRS
Q,CID = C AND CPREFIX = W AND CACTIVITY GT 0
C,ALL,3,$RATIO = NRATIO / (NRATIO + MRATIO)
S CFROM,CTO
P,CFROM,CTO,CACTIVITY,$RATIO
SAV.NODEACTIVITY
Q,(CYEAR = 0 AND CTO = 11 AND CACTIVITY GT 0 AND CFROM GT 0
   AND (CPREFIX = X OR Y)) OR (CYEAR = 1 AND CFROM = 11 AND
    CACTIVITY GT 0)
S,CYEAR
C,CYEAR,$TOTAL = SUM CACTIVITY
P,CPREFIX,CYEAR,CFROM,CTO CACTIVITY,$TOTAL SP 1
SAV.NODEACT-0
Q,(CYEAR = 0 AND CTO = 11 AND CACTIVITY GT 0 AND CFROM GT 0
   AND (CPREFIX = W)) OR (CYEAR = 0 AND CFROM = 11 AND
    CACTIVITY GT 0 AND (CPREFIX = X OR Y))
S,CPREFIX
C,CPREFIX,$TOTAL = SUM CACTIVITY
P,CPREFIX,CYEAR,CFROM,CTO CACTIVITY,$TOTAL SP 1
@END
```

(2) PFCAA.STD-QUERY/CUMSEG. - Contains all the standard MIRADS query sets used for the cumulative data base (CUMSEG) These queries are loaded to the SAVCUMSEG file by the MIRADS program, SAVGEN, in the PFCAA.IMPLEMENT and the PFCAA.IMPLEMENT/TOTAL run-streams.

```
SAV.NODECAP
Q,ID = R AND PREFIX = N AND YEAR GE 0 AND FROM GT 0
   AND GRADE GT 0 AND ACTIVITY EQ *UPLIMIT
C,ALL,$GRD = GRADE / 10
C,ALL,$IN-SPECIALTY = FROM
S,YEAR,GRADE,D,FROM
P,YEAR,$GRD,$IN-SPECIALTY,ACTIVITY,UPLIMIT
SAV.NOTCAPTO
Q,ID = R AND PREFIX = N AND YEAR EQ 0 AND GRADE GT 0
   AND FROM GT 0 AND ACTIVITY NE *UPLIMIT
C,ALL,$IN-SPECIALTY = FROM
C,ALL,$GRD = GRADE / 10
C,ALL,3,$FILLS = ACTIVITY / UPLIMIT
S,YEAR,GRADE,D,FROM
P,YEAR,$GRD,$IN-SPECIALTY,ACTIVITY,UPLIMIT,$FILLS
SAV.CUMNODEACT
Q,(CYEAR = 0 AND CTO = 11 AND CACTIVITY GT 0 AND CFROM GT 0
   AND (CPREFIX = Z)) OR (CYEAR = 1 AND CFROM = 11 AND
   CACTIVITY GT 0)
C,ALL,$GRD = CGRADE / 10
C,CYEAR,$TOTAL = SUM CACTIVITY
S,CYEAR,CGRADE
P,CPREFIX,CYEAR,$GRD,CFROM,CTO,CACTIVITY,$TOTAL SP 1
SAV.CUM-SP-PAIRS
Q,(CID = C AND CPREFIX = W AND CGRADE GT 0 AND CACTIVITY GT 0)
C,ALL,$GRD = CGRADE / 10
C,ALL,3,$RATIO = NRATIO / (NRATIO + MRATIO)
S,CGRADE,D,CFROM,CTO
P,CFROM,CTO,$GRD,CACTIVITY,$RATIO
SAV.CUMNODEACTO
Q,(CYEAR = 0 AND CTO = 11 AND CACTIVITY GT 0 AND CFROM GT 0
   AND (CPREFIX = W)) OR (CYEAR = 0 AND CFROM = 11 AND
    CACTIVITY GT 0 AND (CPREFIX = Z))
C,CPREFIX,$TOTAL = SUM CACTIVITY
C,ALL,$GRD = CGRADE / 10
S,CPREFIX,CGRADE
P,$GRD,CPREFIX,CYEAR,CFROM,CTO CACTIVITY,$TOTAL SP 1
SAV.CUMGRDSUB
Q,ID = R AND PREFIX = N AND ACTIVITY LT *UPLIMIT
C,ALL,$GRD-SUB = UPLIMIT - ACTIVITY
S,YEAR,GRADE,D,FROM
P,YEAR,FROM,$GRD-SUB
SAV.CUMASSIGN-2
Q,CID = C AND CPREFIX = Z AND CYEAR = 0 AND CFROM GT 0
   AND CTO = 11 AND CACTIVITY GT 0
S,CGRADE,D,CFROM
P,CGRADE,CFROM,CTO,CACTIVITY
SAV.CUMASSIGNO-2
Q,CID = C AND CPREFIX = W AND CYEAR = 0 AND CFROM GT 0
   AND CTO = 11 AND CACTIVITY GT 0
S,CGRADE,CFROM
C,ALL,$PAY-GRADE = CGRADE / 10
P,$PAY-GRADE,CFROM,CTO,CACTIVITY
Q,(CID = C AND CPREFIX = W AND (CGRADE LT 50 AND GT 30) AND
   CACTIVITY GT 0) OR (CID = C AND CPREFIX = W AND CFROM NE
   *CTO AND CGRADE LT 40 AND GT 20 AND CACTIVITY GT 0) OR
   (ID = R AND PREFIX = N AND YEAR = 1 AND GRADE LT 30) OR
   (ID = R AND PREFIX = V AND ACTIVITY GT 0 AND GRADE GT 20)
P,CFROM,CTO,CACTIVITY
@END
```

d. MIRADS Data Dictionaries. - The following two runstreams provide the user access to the MIRADS data dictionaries.

(1) PFODSAP.CUR-DICT. - This element contains the MIRADS dictionary cards (see Chapter 4, MIRADS Implementation Manual, for detailed description of card formats) that define the size and characteristics of the row and column data in the current segment data base. This element is @ADD'd for the PFMIRADS.DICGEN program in the PFCAA.IMPLEMENT catalogued runstream.

```
1PCURSEG    000150000014012801
1ICAA          Y
1IODSAS        N
1LI01N  00010001 0014
1M00011ID                IDENTITY
1M00021PREFIX            PREFIX
1M00031YEAR              YEAR
1M00041FROM              FROM SPECIALTY
1M00051TO                TO SPECIALTY
1M00061GRADE             GRADE/SEG
1M00071ACTIVITY          ACTIVITY
1M00081LOWLIMIT          LOWLIMIT
1M00091UPLIMIT           UPLIMIT
1LI02C  00010001 0014
1M00101CID                IDENTITY
1M00111CPREFIX            PREFIX
1M00121CYEAR              YEAR
1M00131CFROM              FROM SPECIALTY
1M00141CTO                TO SPECIALTY
1M00151CGRADE             GRADE/SEG
1M00161CACTIVITY          ACTIVITY
1M00171CLOWLIMIT          LOWLIMIT
1M00181CUPLIMIT           UPLIMIT
1M00191MRATIO           MRATIO
1M00201NRATIO           NRATIO
1M00211TL-M             TOUR LENGTH-M
1M00221TL-N             TOUR LENGTH-N
1M00231ALT1             FIRST ALT SOURCE
1M00241PER1             PERCENT OF 1ST ALT
1M00251PRIMARY          PRIMARY SOURCE
1M00261PERP             PERCENT OF PRIMARY SOURCE
1M00271ALT2             ALT2    SOURCE
1M00281PER2             PERCENT OF 2ND ALTERNATE
```

PFODSAP.CUM-DICT. - This element contains the MIRADS dictionary cards (see Chapter 4, MIRADS Implementation Manual, for detailed description of card formats) that define the size and characteristics of the row and column data in the cumulative segment data base. This element is @ADD'd for the PFMIRADS.DICGEN program in the PFCAA.IMPLEMENT and PFCAA.IMPLEMENT/TOTAL catalogued runstreams.

PFODSAP.CUM-DICT Runstream:

```
IFCUMSEG   000150000014012801
IICAA         Y
IIODSAS       N
IL101R  000100010014
IM0001IID                IDENTITY
IM0002IPREFIX            PREFIX
IM0003IYEAR              YEAR
IM0004IFROM              FROM SPECIALTY
IM0005ITO                TO SPECIALTY
IM0006IGRADE             GRADE/SEG
IM0007IACTIVITY          ACTIVITY
IM0008ILOWLIMIT          LOWLIMIT
IM0009IUPLIMIT           UPLIMIT
IL102C  000100010014
IM0010ICID                IDENTITY
IM0011ICPREFIX            PREFIX
IM0012ICYEAR              YEAR
IM0013ICFROM              FROM SPECIALTY
IM0014ICTO                TO SPECIALTY
IM0015ICGRADE             GRADE/SEG
IM0016ICACTIVITY          ACTIVITY
IM0017ICLOWLIMIT          LOWLIMIT
IM0018ICUPLIMIT           UPLIMIT
IM0019IMRATIO           MRATIO
IM0020INRATIO           NRATIO
IM0021ITL-M             TOUR LENGTH-M
IM0022ITL-N             TOUR LENGTH-N
```

e. MAP Processor Directives. - The following runstreams contain the MAP processor directives needed to MAP the FORTRAN programs used in ODSAS.

(1) PFODSAP.MAP/SACS. - To re-MAP the two programs (SACSPREPRO and SACSCREATE) used to determine the specialty requirements from the PERSACS tape(s).

```
@USE X.,PFODSAP.
@MAP     ,X.SACSPREPRO
IN X.SACSPREPRO
END
@MAP     ,X.SACSCREATE
IN X.SACSCREATE
END
```

(2) PFODSAP.MAP/INITIAL. - To re-MAP the program used to perform the attrition and promotion rates computations and create the master input parameters file (ODINPUDØ2).

```
@USE X.,PFODSAP.
@ASG,A PFRLIBS.
@MAP     ,X.INITIAL
IN PFODSAP.INITIAL
LIB PFRLIBS.
END
```

(3) <u>PFODSAP.MAP/TOURATIOS</u>. - To re-MAP the program used to compute the utilization ratios and tour lengths that make up the specialty preferences file (ODTURUDØ1).

```
@USE X.,PFODSAP.
@ASG,A PFRLIBS.
@MAP     ,X.TOURATIOABS
IN X.TOURATIOS
LIB PFRLIBS.
END
```

(4) <u>PFODSAP.MAP/MATRIX</u>. - To re-MAP the matrix generator program.

```
@USE X.,PFODSAP.
@PREP X.
@ASG,A PFRLIBS.
@MAP     ,X.ABS
NOT TPFS.
SEG BASE
IN X.MAIN
SEG MAIN*,(BASE)
IN X.OPT
IN X.BLOCKDATA
IN X.IBITS
IN X.VALID
IN X.IPHASE
IN X.IPROB
IN X.JPHASE
SEG ONE*,(MAIN)
IN X.INPUT1
SEG TWO*,(MAIN)
IN X.ROWCHP
SEG THREE*,(MAIN)
IN X.RESLO
IN X.ROWOP
SEG FOUR*,(MAIN)
IN X.RESHI
SEG FIVE*,(MAIN)
IN X.LOCOL
IN X.LOCOLC
IN X.LOCOLS
IN X.LOCOLL
SEG SIX*,(MAIN)
IN X.LODIAG
IN X.RES2GR
SEG SEVEN*,(MAIN)
IN X.HICOL
IN X.HICOLS
IN X.CPDIAG
SEG EIGHT*,(MAIN)
IN X.RHS
IN X.RANGE
IN X.BOUNDS
IN X.MASK
SEG SORT1*,(BASE)
IN X.SORTW
IN X.SORTXY
LIB X.
LIB PFRLIBS.
END
```

(5) PFODSAP.MAP/DATABASE. - To re-MAP the programs used to create the data base input files to MIRADS.

```
@USE X.,PFODSAP.
@PREP X.
@MAP   ,X.DATABASE
NOT TPFS.
IN X.DATABASE,.RECORD,.NEWSAVI.COMBIN
IN X.MASKCK
LIB X.
END
@MAP   ,X.DB-CORRECT
IN X.DB-CORRECT
END
```

(6) PFODSAP.MAP/LINKAGE. - To re-MAP the program used to link one system segment to another.

```
@USE X.,PFODSAP.
@MAP      ,X.LINKABS
IN X.LINKAGE
IN X.KTREQ
IN X.OPT
END
```

(7) PFODSAP.MAP/SEPARATE. - To re-MAP the program used to separate the inactive from the active records in the cumulative data base.

```
@USE X.,PFODSAP.
@PREP X.
@MAP   ,X.SEPARATE
IN X.SEPARATE
LIB X.
END
```

(8) PFODSAP.MAP/DBLOAD. - To re-MAP the program used to load the MIRADS data base file, MASfn (where fn is either CUMSEG or CURSEG).

```
@USE X.,PFODSAP.
@ASG,A PFMIRADS.
@MAP   ,X.DBGEN
IN X.DBCREATE
LIB PFMIRADS.
END
```

(9) PFODSAP.MAP/PROCUREMENT. - To re-MAP the two programs used to interface with the MIRADS "HITFILE" of query-selected data base records and produce the ODSAS procurement report.

```
@USE X.,PFODSAP.
@MAP  ,X.HITFILE/INTERFACE
IN X.HITFILE/INTERFACE
LIB PFMIRADS.
END
@ASG,A DAPCPSSA*PF-REL.        . TO USE ALTERNATE PRINT FILE
@MAP   ,X.PROCUREMENT
IN X.PROCUREMENT
IN DAPCPSSA*PF-REL.NTAB
END
```

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER X
DICTIONARY OF PRINCIPAL VARIABLES

1. Purpose. - The purpose of this chapter is to define the principal variables used in all the programs in the Initialization Phase of ODSAS, and in the matrix generator, data base creation and linkage major activities in the Processing Phase.*

2. Data Dictionary for the Initialization Phase. - The name, dimensions (where applicable), and definition of each variable are as follows:

Name	Dimension	Definition
AES	(5Ø)	Advanced entry specialty array
ATRCPT	(15)	Attrition for CPTs by (YOS) for promotees in LT segment
ATRMAJ	(32)	Attrition for MAJs by (YOS) for promotees in CPT segment
ATRTOT		Sum of attrition for 1st and 2d LTs
ATTHI	(6,9)	Weighted average attrition rates for promotees (grade, year). Row 1 not used
ATTLO	(6,9)	Weighted average attrition rates for those remaining in grade (grade, year). Row 1 not used
AUTHMX	(6)	Number of officers authorized by (grade)
BGNYOS		Beginning year of service, used to define earliest year group to be represented
CPT	(15)	CPT population by (YOS) in LT segment

*The source language programs used in the FMPS and on-line inquiry major activities are proprietary software, and as such are not available to the user (thus, variables cannot be defined).

Name	Dimension	Definition
CPTATN		Total attrition for CPTs with over eight YOS (without promotions)
CPTATR		Total attrition for CPTs with over eight YOS (with promotions)
CPTPOP		Total population of CPTs with over eight YOS
CPTREM	(15)	Population of CPTs remaining after attrition, by (YOS)
CPTRM		When used in CPT segment: CPTs remaining rate, for CPTs with less than eight YOS. When used in LT segment: LTs remaining that have not been promoted to CPT
DBLZRO		Counter for the number of unknown specialties
ENDYOS		Ending YOS, used to define oldest year group to be represented
FIVEA		Counter for the number of specialties with five as the first digit, and the second digit either an "A" or unknown
FOURBK		Counter for the number of specialties with four as the first digit and the second digit unknown
GRADE		Index for population and rate arrays
IAUTH		Number of positions authorized from SACS data
IALT		Alternate specialty
IALT1		Index for correct alternate specialty number
ICHG		Used in CPT and LT system segments only. Used for captain's-year when last year group reaches 8th YOS; used for lieutenant's-year when first year group reaches 8th YOS

Name	Dimension	Definition
ICNT		Indicates when there is a need to add population of last year group to previous year group, so that last year group represents ENDYOS and over
IEDATE		Effective date of authorized position, from SACS data (YYMMDD)
IGR		Index for next higher grade
INDEX		Pointer for YOS position in rate arrays
IPNTR		Index for valid specialty number
IPR		Primary specialty
IPR1		Pointer for correct primary specialty number
IRATE	(7,31)	Equivalent to RATE(7,31) for output of integer values stored in matrix
IRAY	(1Ø)	Array for temporary storage of a SACS data record
IREC		Random access file record key
ISPEC		Actual OPMS specialty number
ISUM		Sum of logistics specialty requirements
ITDATE		Termination date of authorized position, from SACS data (YYMMDD)
ITIME	(1Ø)	1Ø one-year time intervals, starting at $T_Ø$
ITREQ	(11,6,1ØØ)	Total requirements by year, grade, specialty
IUTIL	(6)	Integer value for decimal utilization ratios (grade). Not applicable for Grades 3, 2, and 1.
IUTLR8	(11)	Integer utilization ratios used to convert decimal values input

Name	Dimension	Definition
IVAL		Invalid specialty indicator
JCNT		Used for adjusting MAJs ENDYOS population (similar to ICNT)
LTR8W	(2,3Ø)	Attrition rate, with promotion to next higher grade included, by (grade, YOS). Array is for LTs (Grades O-1, O-2) only.
LTR8WO	(2,3Ø)	Attrition rate, without promotion to next higher grade included, by (grade, YOS). Array is for LTs (Grades O-1, O-2) only.
MODE		Identifier of data type: "TEST" for test data or "PROD" for production data; also identifier of whether to create or update TOURATIOS file
MRATIO		Part of utilization ratio that pertains to primary specialty
NAME1		First four characters of FMPS problem name
NAME2		Last four characters of FMPS problem name
NBRAES		Number of advanced entry specialties
NBRPRO		Number of prohibited alternate specialties (in TOURATIOS program only)
NCARD		Number of cards read-in
NPRO		Number of specialties in Segment 1
NPROB	(1Ø)	Prohibited alternate specialty numbers (maximum number of 1Ø)
NPREF		Number of preferences
NRATIO		Part of utilization ratio that pertains to alternate specialty
NSPEC		Number of specialties

Name	Dimension	Definition
NYRS		Number of years to be projected
ONEBLK		Counter for the number of specialties with one as the first digit and the second digit unknown
PMTCPT	(15)	New promotees to CPT (YOS) in the LT segment
PMT1LT	(15)	New promotees to 1LT (YOS) in the LT segment
POP	(6,32)	Population array by (grade, YOS)
POPMAJ		Total new promotees to MAJ
POPSUM		Total population of LTCs or MAJs
POPULA	(1,4)	Accumulators used for population and attrition in Grades 6 through 4 (1,1) total population (1,2) total attrition (1,3) total promoted (1,4) total population less promotions
POP1LT		Total population of 1LTs
POP2LT		Total population of 2LTs
PRMT	(6,9)	Weighted average promotion rate (grade, year); Row 1 not used
PRODIF		Total promotees to MAJ in the CPT (with less than eight YOS) segment
PROMAJ	(32)	New promotees to MAJ in CPT segment, by (YOS)
PSUBTL		Population of CPTs with eight YOS or less at $T_{\emptyset}$

Name	Dimension	Definition
RATE	(7,31)	Array containing rate data per year (for nine years) for all grades, and certain additional data for CPTs and LTs only. Row 1 - for CPTs only: 2Ø AES numbers, nine CPTRM rates, and ICHG Row 2 - for Grade 2: nine rates each for ATTHI, ATTLO, and PRMT; one rate each for OFLOHI, UFLOHI, OFLOLO, and UFLOLO Row 3 - for Grade 3: same as Row 2 Row 4 - for Grade 4: same as Row 2 Row 5 - for Grade 5: same as Row 2 Row 6 - for Grade 6: same as Row 2 Row 7 - for LTs only: same as Row 1
REMMAJ	(32)	MAJs remaining by (YOS) after attrition
RESUM		Population of LTCs or MAJs remaining, after attrition
RSUBTL		Attrition of CPTs with eight YOS or less
R8WOPR	(6,32)	Attrition rate, without promotion to next higher grade included (grade, YOS)
R8WPR	(6,32)	Attrition rate, with promotion to next higher grade included (grade, YOS)
SARRAY	(3,5Ø)	Percent of each alternate specialty's requirements that can be filled in Segment 1 (grade, specialty number)
SEGCOL		Segmentation parameter for COL segment processing (Ø-unsegmented, 1-segmentation within grade)
SEGLTC		Segmentation parameter for LTC segment processing (Ø-unsegmented, 1-segmentation within grade)
SEGMAJ		Segmentation parameter for MAJ segment processing (Ø-unsegmented, 1-segmentation within grade)
SIXA		Counter for the number of specialties with six as the first digit and the second digit an "A"

Name	Dimension	Definition
SIXBLK		Counter for the number of specialties with six as the first digit and the second digit unknown
SPECLT	(5Ø)	Segment 1 specialties
SUM		Total promotions of LTCs or MAJs
TATR1		Total attrition of 2LTs
TATR2		Total attrition of 1LTs
TATR3		Total attrition of CPTs
TATR4		Total attrition of MAJs
THRAB		Counter for the number of specialties with three as the first digit and the second digit an "A" or "B"
THRCD		Counter for the number of specialties with three as the first digit and the second digit a "C" or "D"
THREF		Counter for the number of specialties with three as the first digit and the second digit an "E" or "F"
THRGH		Counter for the number of specialties with three as the first digit and the second digit a "G" or "H"
THRLP		Counter for the number of specialties with three as the first digit and the second digit an "L" or "P"
THRQ		Counter for the number of specialties with three as the first digit and the second digit a "Q"
THRS		Counter for the number of specialties with three as the first digit and the second digit an "S"

Name	Dimension	Definition
THRUV		Counter for the number of specialties with three as the first digit and the second digit a "U" or "V"
THRW		Counter for the number of specialties with three as the first digit and the second digit a "W"
TOTAL		Total population of CPTs with less than eight YOS
TOTATR		Total attrition for promotees
TOTPOP		Total population of CPTs
TOTPRO		Total new promotees to MAJ
TOTREM		Total remaining MAJs
TOUR	(6)	Integer value (i.e., tour length x 1Ø) for primary specialty (grade); not applicable for Grades 3, 2 and 1
TPROM		Total new promotees to MAJ (from CPTs with eight or more YOS only)
TPRMT2		Total promotees to 1LT
TPRMT3		Total promotees to CPT
UPBND	(3,5Ø)	Maximum number that can enter the network within each designated primary specialty
URATIO	(5Ø,5Ø,6)	Utilization ratios and tour lengths by specialty and grade (5Ø,5Ø,1) utilization ratios for Grade 6 (5Ø,5Ø,2) utilization ratios for Grade 5 (5Ø,5Ø,3) utilization ratios for Grade 4 (5Ø,5Ø,4) tour lengths for Grade 6 (5Ø,5Ø,5) tour lengths for Grade 5 (5Ø,5Ø,6) tour lengths for Grade 4 Grades 3, 2, and 1 are not input

Name	Dimension	Definition
UTIL	(6)	Decimal value for utilization ratios (grade). Not applicable for Grades 3, 2, and 1.
UTLR8	(12)	Decimal values which are limits used to convert decimal utilization ratios to integer ratios
VAL	(32)	Attrition, including promotion, of CPTs (YOS)
VALUE		Total attrition in Grades 6 through 4
VAL1	(32)	Attrition, excluding promotion, of CPTs (YOS)
VAL1LT		Attrition of 1LTs
VAL2LT		Attrition of 2LTs
WORK1	(32)	Used for storage of current grade population (YOS)
WORK2	(32)	Used for storage of promotions (YOS)
WORK3	(32)	Used for storage of population less promotions (YOS)

3. Data Dictionary for the Processing Phase. - The variables' definitions for this phase are listed in two sections (and alphabetically within sections).

a. Common Variables section. - Contains the variables which are common to the matrix generator and linkage major activities (major activities 1 and 5, respectively) of the Processing Phase.

b. Variables Used in Data Base Creation Activity. - Contains definitions of variables as described by the title.

Common Variables

Name	Dimension	Definition
AES	(5Ø)	Array of advanced entry specialty numbers
ACTIVY		Number filling requirements
ATTHI	(9)	Attrition rates, per year, for promotees
ATTLO	(9)	Attrition rates, per year, for officers remaining in grade
CPTREM	(9)	Percentage of CPTs with less than eight YOS, remaining in the total CPT population, by year
CRQACT		FMPS solution value for activity of N_ _ _CREQ constraint
CORT		Indicator for "CREQ" or "TREQ" record
IBGN		First word in solution record that has useful data
ICHG		Index used in CPT and LT segments: CPT segment - indicates the year when last group attains 8 YOS LT segment - indicates the year when the first year group attains 8 YOS

Name	Dimension	Definition
ICOUNT		Counter for number of double precision records read in
ICT		Identifier of current record type
IFOR2		Indicator for a Type 2 record; last 12 words of this record must be saved for Type 3 record processing
IFOR3		Indicator for a Type 3 record; last 6 words of this record must be saved for Type 4 record processing
IFOR4		Indicator for a Type 4 record; no words are saved
ILAST		Number of rows or column data sets in last row or column solution file record
INDEX		Index for grade; equals JGRADE in COL segment, otherwise JGRADE+1
IPNTR		Index for ATTLO rate (attrition rates for those remaining in grade)
IRATE	(7,31)	Equivalent to RATE(7,31) for output of integer values stored in RATE array
IROW		First half of row or column name from solution file
IROWX		Second half of row or column name from solution file
ISAVE	(18)	"Save" area for record Types 1, 2, and 3 (number of single precision words)
ISEG		Segmentation option (Ø-no segmentation, 1-Segment 1, 2-Segment 2)
ISFL	(126)	Equivalent to double precision solution file record (SOLFX) for output of integer values
ITEMP	(126)	A temporary storage area for constructing complete records for output

Name	Dimension	Definition
ITOTAL		Sum of requirements for a specialty for all years. If equal to Ø, then specialty number is considered to be invalid
ITREQ	(11,6,1ØØ)	Requirements array (year, grade, specialty)
IVALUE		Word containing XQT options (if other than blank, audit trail output is generated)
JGRADE		Current grade being processed
K1		Index used to get row/column name from solution file
K2		Used to compute the index for activity value, (equals K1+3)
LAST		Number of rows, or column data sets in a solution file record
LYR		Year part (two digits) of the constraint name N_ _TREQ
MAXSPC		Maximum number of specialties
MAXYRS		Maximum number of years
MODE		Identifier of data - "TEST" for test data or "PROD" for production data
MRATIO		Part of utilization ratio that pertains to primary specialty
NAME	(6)	Array of constraint names NAME(1) = CREQ NAME(2) = TREQ NAME(3) = GOZO NAME(4) = LINC NAME(5) = CINC NAME(6) = UBSG
NBRAES		Number of advanced entry specialties
NBRPRO		Number of specialties selected for Segment 1

Name	Dimension	Definition
NBRSPC	(5Ø)	OPMS specialty numbers
NPROB	(5Ø)	Selected specialties for Segment 1
NRATIO		Part of utilization ratio that pertains to alternate specialty
NSPEC		Number of specialties
NSPECL		Actual specialty number used for modifying the correct ITREQ value
NUMBER		Argument in calling KTREQ subroutine 1 = print ITREQ array before modifying requirements 2 = print ITREQ array after modifying requirements
NYRS		Number of years projected
OFLOHI		Overflow percentage - modifies requirements data by allowing more than normal requirements for higher grade
OFLOLO		Overflow percentage - modifies requirements data by allowing more than actual requirements for JGRADE
PRMT	(9)	Promotion rate data per (year)
RATE	(7,31)	(See Paragraph 2, Data Dictionary for Initialization Phase, above)
REQONE		Unfilled requirements for higher grade
REQTOT		Unfilled requirements for higher grade and JGRADE (overflow extracted)
REQTWO		Unfilled requirements for JGRADE
REQ1	(5ØØ)	Unfilled requirements for higher grade, used in right hand side coefficient calculation for CREQ constraints
REQ2	(5ØØ)	Unfilled requirements for current grade (JGRADE), plus unfilled requirement for higher grade (REQ1), used in the right hand side coefficient calculation for TREQ constraints and bounds on XN_ _ arcs

Name	Dimension	Definition
RESFLO	(2Ø84)	Code for flow control constraint upon Y arc. Each word contains 12 three-bit codes meaning: Ø - Initial status 1 - Flow from primary specialty 2 - Flow from alternate specialty 3 - Flow from both specialties 4 - Used only in the LT segment. Indicates that designation of alternate specialty to CPTs with eight YOS occurs at point where CPTs with more than eight YOS would also be present.
ROWCNT		Keeps track of number of rows in problem
RTYPE	(2Ø84)	Code for flow control constraint upon X arc. Each word contains 12 three-bit codes meaning: Ø - Initial status 1 - Equality constraint 2 - Redundant constraint 3 - "Upper Bounded" code
SARRAY	(5Ø)	Percentage of requirements that can be filled with selected specialties during Segment 1
SOLFIL	(6Ø)	Solution file record of 60 double precision words
SOLFX	(63)	Equivalent to SOLFIL but used for output of reconstructed record
SPECLT	(5Ø)	Primary specialties specified for Segment 1
SURVHI	(9)	Survival rate data per year for promotees (1 minus ATTHI)
SURVLO	(9)	Survival rate data per year for those remaining in grade (1 minus ATTLO)
TOUR	(5Ø)	Tour lengths for specialties in JGRADE

Name	Dimension	Definition
UFLOHI		Underflow percentage to allow under-filling a requirement in the higher grade
UFLOLO		Underflow percentage to allow under-filling a requirement in the current grade
ULIM		Maximum number of assignments permitted in a specialty/grade combination
ULIMIT		Maximum number of assignments permitted in a specialty/segment/grade combination
UPBND	(5Ø)	Upper bound on amount of flow for designated specialties
UTIL	(5Ø,5Ø)	Utilization ratios for X arcs; those remaining in grade: 99 - primary/alternate specialty combination not preferred 88 - for m/n specialty pair n is a prohibited alternate Other: a preferred primary/alternate specialty combination
X1		Number attrited - solution value for number assigned to a specialty attrition rate
YTOUR	(5Ø)	Tour length for specialties in higher grade
YUTIL	(5Ø,5Ø)	Utilization ratios for Y arcs; promotees: 99 - primary/alternate specialty combination not preferred 88 - for m/n specialty pair n is a prohibited alternate Other: a preferred primary/alternate specialty combination
ZERO		Ø

Variables Used in Data Base Creation Activity

Name	Dimension	Definition
A	(7)	Array to accept input record A(1) Record type/first character of row or column name A(2) Year/primary specialty number/alternate specialty number/grade A(3) Activity solution value A(4) Lower limit solution value A(5) Upper limit solution value A(6) Number of tours in primary specialty/number of tours in alternate specialty/tour length of primary specialty/tour length of alternate specialty A(7) Segment number
ALPHA		Decoded first three characters of NAME(1) also first character of NAME(1)
BETA		Decoded first four characters of NAME(2)
DELTA		Decoded fourth character of NAME(1)
HDR	(3)	First three double precision words in record. Contains FMPS procedure name, section name (rows or columns), record number and last word indicator
IN	(3)	Equivalent to A(3) through A(5)
INDEX		Equivalent to A(2)
ISUM	(3)	Accumulates A(3) through A(5) for records with same year, specialties, and grade
IVAL	(3)	Contains solution values for activity, lower limit, and upper limit
JINDEX		Equivalent to A(2)
KEY		FLD function of first six characters of NAME(2)

Name	Dimension	Definition
L	(19)	Contains 19 data values indicating the last single precision word that could possibly be used in a record in ODDBSUD1.
NAME	(2)	Row or column name
NEXT	(456)	Integer equivalent of SOLFIL
NREC		Number of row or column records in SOLFIL
SOLFIL	(228)	Contains up to four records (each containing 6Ø double precision words) from ODDBSUD1

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER XI
ESSENTIAL PROGRAM NARRATIVES

1. Purpose. - The purpose of this chapter is to provide narrative information on the ODSAS FORTRAN programs, thereby supplementing the comments within each source program (listed in Chapter XII). The program narratives are arranged by phase, and activity within phase. The following alphabetical index is provided to assist in locating the respective program narratives.

a. Initialization phase programs	Page
INITIAL	XI-7
SACSCREATE	XI-5
SACSEXTRACT	XI-2
SACSPREPRO	XI-3
TOURATIOS	XI-7

b. Processing phase programs	
BOUNDS	XI-32
COMBIN	XI-34
CPDIAG	XI-29
DATABASE	XI-33
DB-CORRECT	XI-40
DBGEN	XI-40
HICOL	XI-28
HICOLS	XI-28
HITFILE/INTERFACE	XI-40
INPUT	XI-8
KEYARC	XI-16
LINKAGE	XI-42
LOCOL	XI-16
LOCOLC	XI-17
LOCOLL	XI-22
LOCOLS	XI-17
LODIAG	XI-24
MAIN	XI-8
MASK	XI-33
MASKCK	XI-34
NEWSAV	XI-33
PROCUREMENT	XI-41

b. Processing phase programs (cont)

	Page
RANGE	XI-31
RECORD	XI-37
RESHI	XI-15
RESLO	XI-14
RHS	XI-31
ROWCHP	XI-9
SEPARATE	XI-45
SORTW	XI-33
SORTXY	XI-33
UPDATE	XI-42

2. Initialization Phase Program Narratives

a. Activity to Create Positional Requirements File

(1) Program: SACSEXTRACT

(a) This program reads the SACS Personnel and Equipment Detail Record File (9-track, 8ØØ BPI tape(s) recorded in EBCDIC, furnished by USAMSSA), extracts officer authorization data for active army units therefrom, and produces a 16ØØ BPI FIELDATA tape for use on System 2 at MILPERCEN. (See Figure III-8 for a generalized graphic portrayal.)

(b) The user must prepare 2 parameter cards (card images)--the first card indicating the number of input tapes and the second card the reel numbers of the tape(s). Entries on the second card must be in A6 format, and up to 6 reel numbers may be entered.

(c) Each input record contains 76 characters and the block size is 95 records. The component field (character number 72) and the officer grade indicator field (characters 2Ø and 21) are validated for each input record. For character 72, only the number "1" (active army) is accepted; for character 2Ø, only the letter "O" (officer) is accepted, and, for character 21, only the numbers "2" through "6" are accepted.

(d) A validity check is also performed on the primary specialty number in each input record. The primary specialty number normally is located in characters 22 and 23. However, if character 24 is non-numeric, then the valid specialty number is located in characters 41 and 42.

(e) Each valid record is written to the output file in the 45-character format shown in Table XI-1.

TABLE XI-1, Format of Valid Output Records Produced by the SACSEXTRACT Program

Print column	Data element	Print column	Data element
1	(blank)	20-25	EDATE
2	Grade	26-31	TDATE
3-4	Primary specialty number	32-40	(blank)
5-19	(blank)	41-45	Number authorized

(2) <u>Program: SACSPREPRO</u>

(a) This program reads the 45-character file created in the SACSEXTRACT Run, decodes the input record into ten fields and stores them in array IRAY(1-10), as shown in Table XI-2.

TABLE XI-2, Storage in Array IRAY(1-10) of Records Decoded by the SACSPREPRO Program

IRAY(i)	SACSEXTRACT record	IRAY(i)	SACSEXTRACT record
i = 1	Grade		TDATE
i = 2	1st position of specialty number	i = 7	YY
		i = 8	MM
i = 3	2d position of specialty number	i = 9	DD
	EDATE	i = 10	Number authorized
i = 4	YY		
i = 5	MM		
i = 6	DD		

(b) The program then edits the primary specialty number field (IRAY(2) and (3)) in each 45-character input record, and writes the edited record to the file ODSAPUD1Ø. For editing purposes, if the second character of the primary specialty number (IRAY(3)) is found to be an integer, the first character is assumed to be an integer also. The primary specialty number is considered valid; otherwise, certain nonstandard specialty numbers (which may appear in the SACS data) are converted to authorized OPMS specialty numbers, as shown in Table XI-3.

TABLE XI-3, Conversion of Certain Nonstandard Specialty Numbers to Authorized OPMS Specialty Numbers by the SACSPREPRO Program

Nonstandard specialty number	Converted to OPMS specialty number	Nonstandard specialty number	Converted to OPMS specialty number
1-	15	3A, 3B	53
2-	15	3C, 3D	52
4-	48	3E, 3F	45
5-	31	3G, 3H	46
5A	11	3L to 3P	99
6-	25	3U, 3V	51
6A	36	3Q	49
3S	97	3W	37

(c) All other nonstandard numbers are considered to be "ØØ." The number of such records is accounted for by the value of the variable DBLZRO.

(d) After all input records have been edited, the identity and number of nonstandard records is printed along with the total number of input records read.

(3) Program: SACSCREATE

(a) This program reads a parameter card which identifies the starting date of the projection period (YYMMDD) and computes the dates for each of the nine years from the starting date, in yearly increments (stored in array ITIME(1-1Ø)). The program then reads a record created in the SACSPREPRO program (record contains grade, primary specialty number, EDATE, TDATE and number authorized) and checks the EDATE and TDATE data fields against the ITIME array. If the EDATE is less than or equal to ITIME(i) and the TDATE is greater than or equal to ITIME(i), the requirements for the primary specialty are counted for year i; otherwise the requirements would not exist at ITIME(i) and thus are not counted for year i. For example, if ITIME(i) equals 77Ø93Ø, and the EDATE is 76Ø93Ø (thus less than or equal to ITIME(i)), and the TDATE is 79Ø93Ø (thus greater than or equal to ITIME(i)), then the number of officers authorized applies in the year beginning at ITIME(i).

(b) A data editing procedure is also employed to detect erroneous specialty numbers in the records. If the specialty number field value does not correspond to its prescribed integer format, the record is erroneous. The record is re-read with an alphanumeric format to ascertain the value in the specialty number field; the erroneous record is written out to the printer for user verification and accounting purposes. The record is bypassed and not counted toward the force structure requirements. If more than 2ØØ records are bypassed, the program branches in order to write out the requirements derived thus far (array ITREQ). Then the program writes out an error message and stops. If the re-read is also erroneous, then an error message is generated and the program immediately branches in a manner similar to that above for bypassing 2ØØ records.

(c) When the editing is completed, the program performs a redistribution of other non-OPMS specialty requirements into authorized OPMS specialties as shown in Table XI-4.

TABLE XI-4, Redistribution of Other Non-OPMS Specialty Requirements into Authorized OPMS Specialties by the SACSCREATE Program

Non-OPMS specialty to be redistributed	Percent of the requirements for this specialty that are to be redistributed to OPMS specialties	OPMS specialty number into which redistributed
Ø7	1ØØ	15
Ø8	1ØØ	42
1Ø	45	11
	18	12
	27	13
	1Ø	14
2Ø	1ØØ	21
22	1ØØ	21
23	1ØØ	21
24	1ØØ	25
34	1ØØ	35
4Ø	75	41
	25	42
98	1ØØ (grades 5 and 6)	7Ø
	1ØØ (grades 2, 3 and 4)	92
99	1ØØ	92
7Ø	a/	71-97

a/ Allocated according to relative strength in each specialty, 71 through 97, after all other redistribution is completed.

b. Activity to Create Preference Files

(1) Program: TOURATIOS

(a) This program operates in two modes, (1) to create a file of utilization ratios, or, (2) to update an existing file of utilization ratios. The first parameter card identifies whether the CREATE mode or the UPDATE mode is desired. The difference between the two modes is that, in the CREATE mode, the ratios are initialized at 99 (meaning no preference) for non-combat arms specialties, and 88 (a unique number indicating a combat arms specialty) for combat arms specialties, whereas for the UPDATE mode, the ratio initialization is bypassed. The second parameter card identifies the specialties that are prohibited as alternate specialties (i.e., combat arms specialties).

(b) The program then reads from a previously created card-image file (ODTURUDØ1) the preferred specialty pairing (e.g., Infantry/Personnel Management would be entered as 1142) followed by decimal utilization ratios and tour lengths of the primary specialty for the grades of COL, LTC and MAJ respectively. The decimal utilization ratios are then converted to a ratio of two integer values. This is done by first determining how the decimal ratio relates to a set of decimal values (in the UTLR8 array)--e.g., the program asks, "Is it greater than the first value, but less than the second?" Then the above relationship can be equated to a predefined integer number (in the IUTLR8 array) identifying the ratio (e.g., an input decimal ratio of .12 is in the range of Ø.Ø to Ø.18 which equates to the integer 15, which, in turn, means 1 tour in the primary specialty followed by 5 tours in the alternate specialty).

(c) For each input of a preferred specialty pair, a utilization ratio for the specialty pair in reverse order is also produced (i.e., the input primary specialty becomes the alternate specialty). For the reverse order specialty pair, the utilization ratios are the complements (e.g., an input decimal ratio of .12 for specialty pair 12/42, becomes (1.Ø - .12 = Ø.88) for specialty pair 42/12).

c. Activity to Compute Attrition/Promotion Rates and Create Input Parameters File

(1) Program: INITIAL

(a) This program reads the ODR8SUDØ1 file of parameters, populations and attrition rates for a reference population, performs all calculations needed to produce the attrition/promotion

rates used in all the system segments, and creates the input file for processing the first COL run. The rate computations are explained in detail in Appendix D of the ODSAS Study Report (reference 4).

(b) Any specialty number input is edited to insure conformance with the authorized set of OPMS specialty numbers (stored in the ISPEC array). In addition, the parameters shown in Table XI-5 are edited for the range of values indicated.

TABLE XI-5, Parameters Edited by the INITIAL Program

Parameter	Range of acceptable values
Number of specialties	1 - 5Ø
Number of years in projected period	1 - 9
Number of Advanced Entry Specialties	1 - 2Ø

3. Processing Phase Program Narratives

a. Matrix Generator Activity

(1) Program: MAIN. - This is the "executive" program in the matrix generator activity which specifies the calling sequence of the subroutines.

(2) Program: INPUT

(a) This program reads the input parameter file (ODINPUDØ1) created for the current grade segment, the specialty preferences file (ODRATUDØ1), and the specialty requirements file (ODSACUDØ1). The problem size parameters, authorized strengths by grade, identification of specialties to have upper bounds on total strength, the first set of segmentation parameters (segmentation code, followed by upper bound information for each specialty identified in card 3), and percentage fill allowed in segment 1 are read from the input parameters file.

(b) Continuation, or survival rates (1.Ø - attrition rate), are computed for use in aging the flows through the network. SURVHI(j) is the survival rate used for Y arc flows in year j, and SURVLO(j) is the survival rate used for X arc flows in year j.

(c) The utilization ratios and tour lengths are extracted from the specialty preferences file (ODRATUDØ1) for the current and next higher grade. The utilization ratios and tour lengths for majors are also used in the captain and lieutenant segments for those flows representing dual qualified officers.

(d) The force structure requirements for the current grade and the next higher grade are then selected from the specialty requirements file. The REQ1 array holds the requirements data for the next higher grade above the current segment grade (if the next higher grade is greater than 6, REQ1 is blank), and initially the REQ2 array holds the requirements data for the current grade. The REQ1 array is then added to the REQ2 array so that the REQ2 array contains the total requirements (current grade plus next higher grade). REQ1 remains with the next higher grade requirements data.

(3) Program: ROWCHP

(a) This program produces the type and name of all constraints (except for the flow control constraints upon flows between different specialties). The type and name selected, however, depend upon the grade and year in the projection period.

(b) The criteria governing selection of the constraint types and names for Grades 6, 5 and 4 in the years in the projection period are as shown in Table XI-6.

(c) The sets of constraints described in Table XI-6 are produced for each year, for each OPMS specialty. However, when the segmentation within-grade option is in effect, the IPHASE subroutine is called to determine whether the processing is for Segment 1 or Segment 2. If for Segment 1, then all specialties are included. If the processing is for Segment 2, then Segment 1 primary specialties are excluded from consideration.

TABLE XI-6, Criteria Governing Selection by the ROWCHP Program of Year-Dependent Constraint Types and Names for Grades 6, 5 and 4

Year							
$T'_{\emptyset}$		$T_{\emptyset}$		T_1-T_{NYRS-1}		T_{NYRS}	
Constraint Type a/	Constraint Name b/	Constraint Type a/	Constraint Name b/	Constraint Type a/	Constraint Name b/	Constraint Type a/	Constraint Name b/
E	WØ_ _GOZO	L	NØ_ _TREQ	L	Nn_ _CREQ c/	L	Nn_ _CREQ c/
		E	NØ_ _GOZO	L	Nn_ _TREQ	E	Nn_ _GOZO
		E	NØ_ _LINC	E	Nn_ _GOZO		
				E	Nn_ _LINC		
				E	Nn_ _CINC c/		

a/ E indicates "equal to"; L indicates "less than or equal to".
b/ The third and fourth positions in the name (indicated by _ _) are for the specialty number.
c/ Applies to Grades 5 and 4 only.

(d) Two additional constraints, independent of year and segmentation option, are also constructed for Grades 6, 5 and 4. These constraints are as shown in Table XI-7.

TABLE XI-7, Additional Constraint Types and Names Produced by the ROWCHP Program for Grades 6, 5, and 4

Constraint Type[a]	Constraint Name	Description/comment
L	WØ_ _UBSG	the third and fourth positions (indicated by _ _) are specialty numbers having specific upper bounds on the number of officers to be assigned that specialty
L	TOTAUTH	total authorized strength for the grade

[a] L indicates "less than or equal to"

(e) The criteria governing selection of the constraint types and names for Grade 3 in the years in the projection period are as shown in Table XI-8.

(f) The sets of constraints described in Table XI-8 are produced for each year, for each OPMS specialty, except as noted for WØ_ _LINC at $T_Ø$.

(g) The TOTAUTH constraint is also constructed for Grade 3, in the same manner as described for higher grades in (d) above. However, WØ_ _UBSG constraints are not constructed for Grade 3.

TABLE XI-8, Criteria Governing Selection, by the ROWCHP Program, of Year-Dependent Constraint Types and Names for Grade 3

Year					
$T'_{\emptyset}$		$T_{\emptyset}-T_{NYRS-1}$		T_{NYRS}	
Constraint Type[a/]	Constraint Name[b/]	Constraint Type[a/]	Constraint Name[b/]	Constraint Type[a/]	Constraint Name[b/]
E	WØ_ _GOZO	L	Nn_ _TREQ	E	Nn_ _GOZO
E	WØ_ _LINC[c/]	E	Nn_ _GOZO		
		E	Nn_ _LINC[d/]		
		E	Nn_ _CINC		

[a/]E indicates "equal to"; L indicates "less than or equal to".

[b/]The third and fourth positions in the name (indicated by _ _) are for the specialty number.

[c/]A constraint is constructed at $T'_{\emptyset}$ for BES only.

[d/]In the CPT segment, the variable ICHG has the value of the last year when there are CPTs with 8 YOS or less. Therefore, if the year is greater than ICHG, and thus no CPTs with 8 YOS or less, Nn_ _LINC constraints are not cconstructed.

(h) The criteria governing selection of the constraint types and names for Grade 2 in the years in the projection period are as shown in Table XI-9.

TABLE XI-9, Criteria Governing Selection by the ROWCHP Program of Year-Dependent Constraint Types and Names for Grade 2

Year			
$T_{\emptyset} - T_{NYRS-1}$		T_{NYRS}	
Constraint Type[a/]	Constraint name[b/]	Constraint Type[a/]	Constraint name[b/]
L	Nn_ _CREQ[c/]	L	Nn_ _CREQ[c/]
E	Nn_ _GOZO	E	Nn_ _GOZO
E	Nn_ _LINC		
E	Nn_ _CINC[d/]		

[a/] E indicates "equal to"; L indicates "less than or equal to".

[b/] The third and fourth positions in the name (indicated by _ _) are for the specialty number.

[c/] In the LT segment, the variable ICHG has the value of the first year when some LTs would attain 8 YOS. Therefore, only when the year is greater than ICHG will Nn_ _CREQ constraints be constructed.

[d/] If the year is greater than or equal to ICHG, then Nn_ _CINC constraints are constructed.

(i) The sets of constraints described in Table XI-9 are produced for each year, for each OPMS specialty.

(j) The TOTAUTH constraint is also constructed for Grade 2, in the same manner as described for higher grades in (d) and (g) above. However, WØ_ _UBSG constraints are not constructed for Grade 2.

(4) Program: RESLO

(a) This program computes code numbers ("Ø" or "1") identifying which X arcs (for Grades 6, 5, and 4) have flow in them and stores the codes, in a packed form, in the RTYPE array (except as described in (b) below). Given the tour length for specialty n and the utilization ratio for each preferred specialty pair that starts at $T_Ø$ in Specialty n, the number of years when assignments will be made from Specialty n are computed first. Then, for each of those years, a number of arcs are identified for each preferred specialty pair (the variable INDEX = ((YEAR x NSPEC^2) + (N-1) x (NSPEC) + M)) defines the arc's relative position in the network), where officers with a particular specialty pair would be assigned. For example, if some officers with Specialties 11 and 49 leave Specialty 11 at $T_Ø$ and others leave at T_1 (with Specialty 49 being a two-year specialty), the paths (arcs) would be as shown in Figure XI-1.

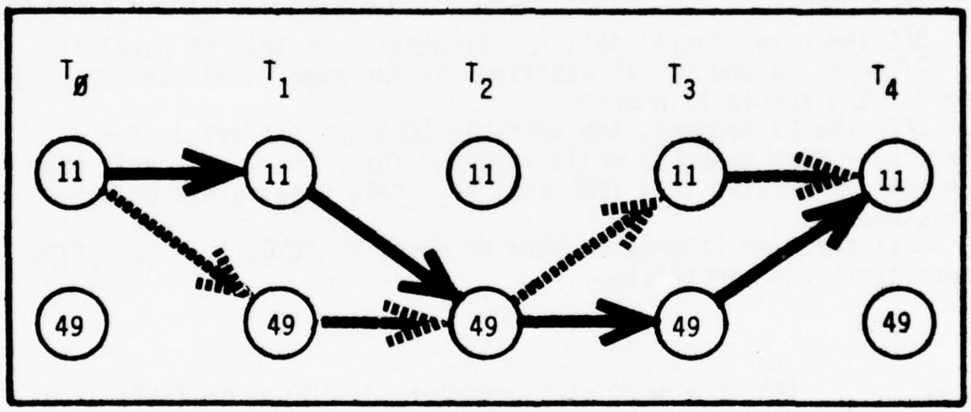


FIGURE XI-1, Example of Arcs Identifiable as Containing Flow and Thus Assigned a Code Number by the RESLO Program (for Grades 6, 5 and 4).

In this example, the identified arcs (both solid and dashed lines) would all receive a code number "1", indicating flow; all other arcs, being without flow, would receive a code number "Ø" at this point in the matrix generator activity.

(b) For Grades 5 and 4, code numbers are also generated to indicate when promotees (to Grades 6 and 5, respectively) would transfer from one specialty to the other. Promotions can occur at any node, and the flow is redirected to a Y arc; however, the flow representing promotees only moves to the other specialty at the end of a tour length. If a promotion occurs during a tour length, the promotee remains assigned to the specialty in which he is serving (i.e., tour length takes precedence over reassignment required because of promotion). The codes so generated are stored, in packed form, in the RESFLO array. The code numbers are as shown in Table XI-10.

TABLE XI-10. Code Numbers Generated by the RESLO Program to Indicate When Promotees in Grades 5 and 4 Transit From One Specialty to Another

Code Number	Meaning
1	Flow of promotees from primary specialty to alternate specialty
2	Flow of promotees from alternate specialty to primary specialty
3	Flow of promotees where the "from" specialty is both a primary and an alternate

(5) Program: RESHI

(a) This program computes the codes identifying which Y arcs (for Grades 5, 4, 3, and 2) have flow in them and the classification of the flow (i.e., flow from an officer's primary, or alternate, to an alternate or primary, respectively, or both). The codes are stored in a packed form in the RESFLO array.

(b) For Grades 5 & 4, the arcs which represent the path segments where promotees move to their second specialty for the first time were coded in the RESLO subroutine. Each of those arcs is treated in the RESHI subroutine as the beginning of a path for promotees and the remainder of the path is logically constructed and coded to indicate the classification of flow in all the path segments (in a manner similar to that described in subroutine RESLO). Codes of 1, 2 or 3 are used, as in RESLO, and have the same meanings.

(c) For Grade 3, the codes identifying promotions (actually, designations of alternate specialties at the eighth YOS point) are first generated in this subroutine. For each BES, starting at year Ø and continuing up through the last year where CPTs will be designated alternate specialties (value of the variable ICHG), the arc leaving the BES and going to a preferred alternate is coded as a "1" (promotion from primary specialty). Each of the arcs coded with a "1" is then treated as the beginning of a path (similar to treatment of Grades 5 & 4 described in (b) above), and the remainder of the path is logically constructed and coded to indicate the classification of flow in all the path segments (as described for subroutine RESLO).

(d) The codes for the flows representing CPT/MAJ with greater than eight YOS (Y arc flows) are then computed in the same manner as described in RESLO for X arcs. The codes for this group are also stored in the RESFLO array.

(e) For Grade 2, the codes identifying promotions (actually designation of alternate specialties at the eighth YOS point) are generated for arcs beginning at the year defined by the variable ICHG. During the years commencing at ICHG and continuing to NYRS, the flows in the Y arcs are not restricted (rather the requirements draw the available flow) unless earlier designees are due to return to their other specialty. In this latter case, the restrictive flow constraint is a "less than or equal to" type, since designations are also occurring in that year and they are requirements driven, and therefore not subject to an equality constraint.

(6) Program: KEYARC. - This program constructs the key arc relationship constraints for grades 6 through 3 (the LT segment does not utilize this type constraint). A "less than or equal to" type constraint is constructed for each preferred specialty pair where the primary specialty number is less than the alternate specialty number. For instance, the constraint, UR2142, relates the flow in the arcs WØ4221 and WØ2142.

(7) Program: LOCOL. - This program computes the number of preferences that apply to the grade segment or subsegment to be processed. The number of constraints by type and the total number of constraints are then computed and those values are then included in the matrix generator statistical summary report. Lastly, depending upon the grade segment to be processed, the appropriate subroutine--LOCOLS (for Grades 6, 5, and 4), LOCOLC (for Grade 3), or LOCOLL (for Grade 2)--is called to define variables for the FMPS COLUMN chapter input.

(8) Program: LOCOLS. - This program has three major functions, described in Subparagraphs (a) (b) and (c) below, respectively.

(a) The first function is to define variables for the FMPS COLUMN chapter input for grades 6, 5 and 4. These variables, with the several distinct forms for their names and the coefficients associated therewith are as shown in Table XI-11.

(b) The second function of LOCOLS is to produce a record for input to the current segment data base for each WØmn, and Xjmn variable. Each WØmn record contains the variable name, the utilization ratio, and the tour lengths of specialties m and n. Each Xjmn variable contains the variable name, the utilization ratio, the tour lengths of specialties m and n, and the identification of an X and/or Y arc in the path starting with Xjmn.

(c) The third function of LOCOLS is to produce that part of the optional matrix generator report which indicates the last exit year points for officers with specialties m and n. (See Chapter V--paragraph 3a(2)(c) and Figure V-9--for a detailed description of the report.)

(9) Program: LOCOLC. - This program has two major functions, described in Subparagraphs (a) and (b), below, respectively.

(a) The first function is to define variables for the FMPS COLUMN chapter input for Grade 3. These variables, with the several distinct forms for their names, and the coefficients associated therewith are as shown in Table XI-12.

(b) The second function of LOCOLC is to produce a record for input to the current segment data base for each WØmn variable shown in Table XI-12. Each record contains the variable name, the utilization ratio, the tour lengths of specialties m and n, the identification of the Y arc fed by WØmn, and the percentage of WØmn that feeds the Y arc.

TABLE XI-11, Variable Names and Coefficients (for Grades 6, 5 and 4) Defined by the LOCOLS Program for FMPS COLUMN Chapter Input (continued on next page)

Variable	Grade	Constraint Names	Constraint Coefficients
XØØØn	6,5,4	WØn GOZO	+1.ØØØ
		WØn UBSG	+1.ØØØ (if specialty n has an upper bound)
WØmn	6	WØmGOZO	-1.ØØØ
		WØnGOZO	+1.ØØØ
		NØnTREQ	+1.ØØØ
		TOTAUTH	+1.ØØØ
		WØnUBSG	+1.ØØØ (if specialty n has an upper bound)
		OBJECTIV	+1.ØØØ
		UR i j (i=min(m,n), j=max(m,n))	if m<n, left member of integer utilization ratio is applicable to specialty pair m/n; if m>n, (-1.Ø x left member of integer utilization ratio) is applicable to specialty pair m/n
		NØn LINC	(1.Ø-PRMT(1)) x (fraction remaining in specialty n)
		Njn LINC	coefficient calculation considers all of the following: 1) a fraction of the WØmn population remains in specialty n, 2) the remaining population is subject to attrition, and 3) a percentage of the WØmn population can get promoted (promotion rate in the COL segment is Ø.), in each year
		RØnm	(1.Ø-PRMT(1)) x (fraction departing specialty n)
		Rjnm j>Ø	for each year j>Ø (where j is a year when the $T_Ø$ population would initially depart specialty n) the coefficient calculation considers all of the following: 1) a fraction of the WØmn population departs specialty n, 2) the WØmn population is subject to attrition until departing, and 3) a percentage of the WØmn population can get promoted (promotion rate in the COL segment is Ø.)

TABLE XI-11, Variable Names and Coefficients (for Grades 6, 5 and 4) Defined by the LOCOLS Program for FMPS COLUMN Chapter Input (concluded)

Variable	Grade	Constraint	
		Names	Coefficients
WØmn (cont)	5,4	Same as grade 6, plus the following:	
		RESØmn	PRMT(I)
		RESjmn	for each year j>Ø (where J is a year when the $T_Ø$ population would initially depart specialty n) the coefficient calculation considers all of the following: 1) a fraction of the WØmn population remains in specialty n, 2) the remaining population is subject to attrition 3) a percentage of the WØmn population can get promoted in each year
Xjmn j>Ø m=n	6,5,4	NjmGOZO	-1.Ø
		N(j+1)GOZO	SURVLO (j+1)
		NjmLINC	-1.Ø
		N(j+1)mTREQ	+1.Ø (omitted if (j+1) = NYRS)

TABLE XI-12, Variable Names and Coefficients (for Grade 3) Defined by the LOCOLC Program for FMPS COLUMN Chapter Input (continued on next page)

Variable	Constraint Names	Constraint Coefficients
XØØØn	WØnGOZO	+1.ØØØ (only if n is a BES)
	WØnLINC	CPTREM(1)
	NØnLINC	CPTREM(2)
	NjnLINC[a/] j>Ø	CPTREM(j+2)
WØmn (m=n and m is a BES)	WØmGOZO	-1.ØØØ
	NØnGOZO	+1.ØØØ
	WØmLINC	-1.ØØØ
	NØnTREQ	+1.ØØØ
	TOTAUTH	+1.ØØØ
Xjmn (m=n, Ø<j<ICHG, and m is a BES)	NjmGOZO	-1.ØØØ
	N(j+1)nGOZO	SURVLO(j+1)
	NjmLINC	-1.ØØØ
	N(j+1)nTREQ	+1.ØØØ (if (j+1) ≤NYRS)

TABLE XI-12, Variable Names and Coefficients (for Grade 3) Defined by the LOCOLC Program for FMPS COLUMN Chapter Input (concluded)

Variable	Constraint	
	Names	Coefficients
Yjmn (m=n and Ø<j<NYRS)	NjmGOZO	-1.ØØØ
	N(j+1)nGOZO	SURVHI(j+1)
	NjmCINC	-1.ØØØ
	N(j+1)nTREQ	+1.ØØØ (if (j+1) ≤NYRS)
WØmn (m≠n)	WØmGOZO	-1.ØØØ
	NØnGOZO	+1.ØØØ
	NØnTREQ	+1.ØØØ
	TOTAUTH	+1.ØØØ
	UR i j (i = min(m,n), j = max(m,n))	if m<n, left member of integer utilization ratio is applicable to specialty pair m/n; if m>n, (-1.Ø x left member of integer utilization ratio) is applicable to specialty pair m/n
	NjnCINC	coefficient calculation considers both of the following: 1) a fraction of the WØmn population remains in specialty n at year j, and 2) the remaining population is subject to attrition
	RESjnm	coefficient calculation considers the fraction of the WØmn population leaving specialty n at year j in addition to the two factors considered for NjnCINC, above

a/ Constructed as long as there is a CPT population with less than 8 YOS

(10) Program: LOCOLL. - This program has two major functions, described in subparagraphs (a) and (b), below, respectively.

(a) The first function is to define variables for the FMPS COLUMN chapter input for Grade 2. These variables, with the several distinct forms for their names, and the coefficients associated therewith are as shown in Table XI-13.

(b) The second function of LOCOLL is to produce a record for input to the current segment data base for each XØØØn and Xjmn(m=n) variable shown in Table XI-13. Each record contains the source variable name, the identification of an X or Y arc in the path, and the percentage of the source variable that feeds the X or Y arc.

TABLE XI-13, Variable Names and Coefficients (for Grade 2) Defined by the LOCOLL Program for FMPS COLUMN Chapter Input

Variable	Constraint	
	Names	Coefficients
XØØØn	WØnGOZO	+1.ØØØ
	NØnLINC	(1.Ø-PRMT(1))
	TOTAUTH	+1.ØØØ
Xjmn (m=n, and j≥Ø)	NjmGOZO	-1.ØØØ
	N(j+1)nGOZO	SURVLO(j+1)
	NjmLINC	-1.ØØØ
	N(j+1)nTREQ	+1.ØØØ
	N(j+1)nLINC	(SURVLO(j+1)) x (1.Ø-PRMT(j+2))
	N(j+1)nCINC[a]	(SURVLO(j+1)) x (PRMT(j+2))
Yjmn (m=n, and j≥Ø but <NYRS)	NjnGOZO	-1.ØØØ
	N(j+1)nGOZO	SURVHI(j+1)
	N(j+1)nTREQ	+1.ØØØ
	NjnCINC[b]	-1.ØØØ
	N(j+1)nCREQ[b]	+1.ØØØ
	N(j+1)nCINC[b]	(SURVHI(j+1)) x (CPTREM(j+1))

a/only defined if (j+1) is ≥ ICHG (year when designations begin
b/only defined if j≥ ICHG

(11) Program: LODIAG. - This program defines Xjmn ($j \geq \emptyset$ and $m \neq n$) variables and their coefficients in the TREQ, GOZO, LINC, and flow control (R prefix) type constraints for Grades 6, 5 and 4.

(a) The flow in each X arc, representing officers departing a Specialty n for the first time since $T_{\emptyset}$, is traced along a path defined by the grade, utilization ratio and tour length. The general procedure for an X arc is as illustrated for $X_{\emptyset 1149}$ in Figure XI-2.

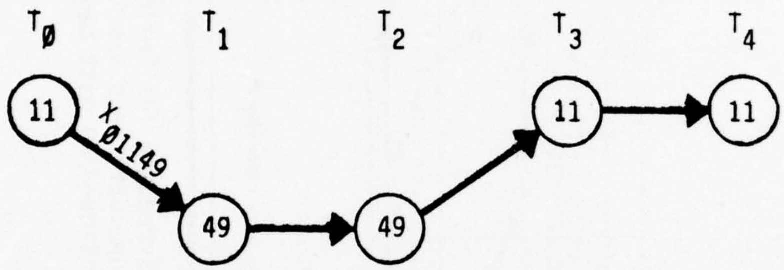
(b) The flow in each of the arcs described above also includes that portion of the flow representing officers due for promotion at each node along the path. The path for this latter portion is traced to the first point in the path of the X arc where a Yjmn ($j \geq \emptyset$ and $m \neq n$) arc would originate.

1. Since the promotion of colonels (Grade 6) is not specifically modeled in ODSAS (such promotion is considered as a loss from the colonel population), the example in Figure XI-2 is fully descriptive of the X arcs in the COL segment.

2. In the LTC and MAJ segments, however, the promotion consideration is applicable. The path of promotees depends upon the tour length of the specialty at the point of promotion. If promotion and tour length completion dates coincide, promotees move to the alternate specialty. Otherwise, they remain in the specialty of current assignment until one tour length has been completed. For example, the promotees in Specialty 49 at T_1 would continue in Specialty 49 until T_2, when they would be reassigned to Specialty 11; similarly, those promoted in Specialty 49 at T_2 would be reassigned to Specialty 11 at that time since the 2-year tour length for officers in $X_{\emptyset 1149}$ is completed at T_2. The promotees are described by a fraction of the original X arc through the point where the first flow control (RES type) constraint for a Y arc is encountered. The RES2GR subroutine of LODIAG computes the value of that fraction of the X arc to be assigned to the Y arc. This procedure is explained in more detail in Figure XI-3.

1. __Assume:__ grade of colonel,
   tour length is 2 years in both Specialties 11 and 49,
   utilization ratio = 1.1,
   NYRS = 4, and
   survival rate (1.Ø - attrition rate) for colonels is
   9Ø percent in each year.

2. __Then:__ the path for an officer, with Specialty pair 11/49, departing Specialty 11 for the first time since $T_Ø$ would be:



3. __And the following constraint names and coefficients would be defined by the LODIAG program for the $X_{Ø1149}$ variables:__

Constraint names	Coefficients
NØ11GOZO	-1.ØØØ
RØ1149	-1.ØØØ
N149GOZO	+Ø.9ØØ
N149TREQ	+1.ØØØ
N149LINC	+Ø.9ØØ
R24911	+Ø.81Ø = (Ø.9) x (Ø.9)
N311LINC	+Ø.72Ø = (Ø.9) x (Ø.9) x (Ø.9)

FIGURE XI-2, Illustrative Example of the General Procedure in LODIAG Program for Defining X Arc Variables and Coefficients for FMPS COLUMN Chapter Input

1. Assume: a starting population of 200 at T_0,
   tour length is 2 years in both specialty 11 and specialty 49,
   NYRS = 4,
   promotion rate each year = .10,
   survival rate in present grade = .90, and
   survival rate in higher grade = .95

2. Then, of a starting population of 200 LTCs (in X_{01149}) at T_0, 180 "survive" (.90 x 200) as LTCs, and all 180 are in specialty 49 at T_1

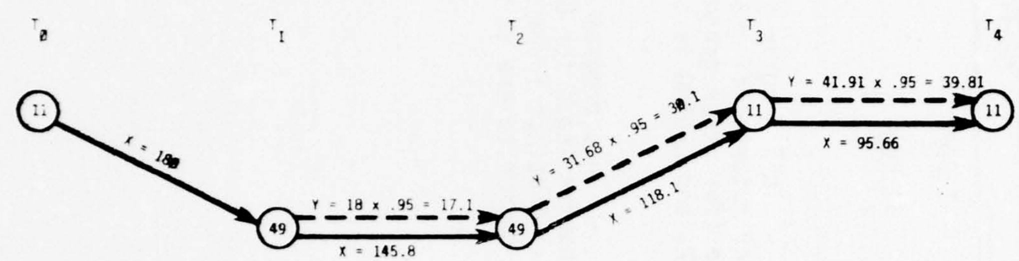
3. The X and Y arc values, derived as shown below, are:

	$T_1 - T_2$ Interval				$T_2 - T_3$ Interval				$T_3 - T_4$ Interval				T_4
	Starting population	Promotion rate	Resulting population	Survival rate	Starting population	Promotion rate	Resulting population	Survival rate	Starting population	Promotion rate	Resulting population	Survival rate	Starting population
Y arcs			18[a]	.95	17.1		(17.1) (14.58) 31.68[a]	.95	30.1		(30.1) (11.81) 41.91[a]	.95	39.81
X arcs	180	.10	162	.90	145.8	.10	131.22	.90	118.1	.10	106.29	.90	95.66
				TOUR LENGTH ENDS	TOUR LENGTH BEGINS							TOUR LENGTH ENDS	TOUR LENGTH BEGINS

[a] Subroutine *RES2GR* (a subroutine within *LODIAG*) computes the input to Y arcs destined to leave a specialty at the end of a tour length and the input to Y arcs that remain in a specialty before the end of a tour length.

FIGURE XI-3, Illustrative Example of the LODIAG Program Procedure for Defining X and Y Arc Variables and Coefficients When Promotion Considerations are Included (Grades 5 and 4), for FMPS COLUMN Chapter Input (continued on next page)

4. The paths for the above X and Y arcs are:



5. And the following constraint names and coefficients would be defined by the LODIAG program:

Constraint name	Coefficient	
N149CINC	0.090	-- ((0.90) x (0.10))
RES24911	0.086	-- ((0.90) x (0.10) x (0.95)) remaining promotees from T_1
	+0.059	-- ((0.90) x (0.90) x (0.90) x (0.10)) promotees at T_2
	0.145	-- Total promotions from X_{01149} at T_2
N311CINC	0.059	-- (0.90) x (0.90) x (0.90) x (0.90) x (0.90) x (0.10)

FIGURE XI-3, Illustrative Example of the LODIAG Program Procedure for Defining X and Y Arc Variables and Coefficients When Promotion Considerations are Included (Grades 5 and 4), for FMPS COLUMN Chapter Input (concluded)

(c). In addition to the functions of defining paths for X arcs and inputs for Y arcs described respectively in (a) and (b) above, LODIAG also defines the XNn variables for FMPS COLUMN chapter input as shown in Table XI-14.

TABLE XI-14, XNn Variable Names and Coefficients Defined by the LODIAG Program for FMPS COLUMN Chapter Input

Variable	Constraint	
	Names	Coefficients
XNn	N(NYRS)nGOZO	-1.000
	OBJECTIV	+1.000

(12) Program: HICOL. - This program tests the grade segment identification and either returns (if grade equals 6), or calls the appropriate subroutine (for other grades). HICOLS is the subroutine for Grades 5, 4 and 2, and CPDIAG is the subroutine for Grade 3.

(13) Program: HICOLS. - This program defines certain variables for the FMPS COLUMN chapter input for Grades 5, 4 and 2. These variables, with the several distinct forms for their names and the coefficients associated therewith are as shown in Table XI-15.

TABLE XI-15, Variable Names and Coefficients Defined by the HICOLS Program for FMPS COLUMN Chapter Input

Variable	Grade	Constraint Names	Constraint Coefficients
Yjmn ($j \geq 1$, m=n)	4,5	NjmnGOZO	-1.000
		N(j+1)nGOZO	SURVHI(j+1)
		NjmCINC	-1.000
		N(j+1)nCREQ	+1.000
		N(j+1)nTREQ	+1.000 (defined only if (j+1) $\neq$ NYRS)
Yjmn (j>0 for grades 4 and 5, m$\neq$n; j>ICHG for grade 2, m$\neq$n)	2,4,5	NjmGOZO	-1.000
		N(j+1)nGOZO	SURVHI(j+1)
		RESjmn	-1.000
		N(j+1)nCREQ	+1.000
		N(j+1)nTREQ	+1.000
Yjmn ($j \geq 0$, m$\neq$n)	4,5	In addition to above, compute path similar to that of the general procedure for X arcs described in LODIAG, i.e., by defining the NjnCINC, RESjnm and RESjmn constraints and their coefficients	
XNn	2	N(NYRS)nGOZO	-1.000
		OBJECTIV	+1.000

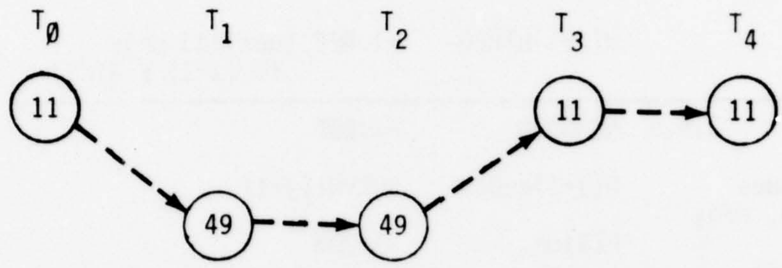
(14) Program: CPDIAG. - This program has two functions, described in Subparagraphs (a) and (b) below, respectively.

(a) The first function is to define the Yjmn ($j \geq 0$ and m$\neq$n) variables and their coefficients in the TREQ, GOZO, CINC and flow control (RES-prefix) type constraints for Grade 3.

The flow in each Y arc representing officers (CPT/MAJ with greater than or equal to 8 YOS) departing a specialty m for specialty n for the first time since $T_Ø$ is traced along a path defined by the utilization ratio and tour length for Grade 4. The general procedure for each Y arc is as illustrated in Figure XI-4.

1. <u>Assume</u>: Tour lengths of 2 years in both Specialties 11 and 49,
   NYRS = 4, and
   Survival rate in current grade (1.Ø - attrition rate) is 9Ø percent in each year.

2. <u>Then the paths for the Y arcs are</u>:



3. <u>And the constraint names and coefficients defined by CPDIAG would be</u>:

Constraint names	Coefficients
NØ11GOZO	-1.ØØØ
RESØ1149	-1.ØØØ
N149GOZO	+Ø.9ØØ
N149TREQ	+1.ØØØ
N149CINC	+Ø.9ØØ
RES24911	+Ø.81Ø = (Ø.9) x (Ø.9)
N311CINC	+Ø.72Ø = (Ø.9) x (Ø.9) x (Ø.9)

FIGURE XI-4, Illustrative Example of the CPDIAG Program Procedure for Defining Y Arc Constraint Names and Coefficients (Grade 3) for FMPS COLUMN Chapter Input

(b) The second function of CPDIAG is to define the XNn variables for the Grade 3 segment. The constraint names and coefficients are as shown in Table XI-16.

TABLE XI-16, Constraint Names and Coefficients (for Grade 3) Defined by the CPDIAG Program for FMPS COLUMN Chapter Input

Variable	Grade	Constraint	
		Names	Coefficients
XNn	3	N(NYRS)nGOZO	-1.ØØØ
		OBJECTIV	+1.ØØØ

(15) Program: RHS. - This program defines the nonzero b-coefficients for the FMPS RIGHT-HAND SIDE chapter input for all grades for the constraints shown in Table XI-17.

TABLE XI-17, Nonzero b-coefficients Defined by the RHS Program for FMPS RIGHT-HAND SIDE Chapter Input

Constraint names	Grade	Coefficients [a]
NjmCREQ	4,5	REQ1(INDEX), where INDEX = ((k-1) x (NYRS+1)) + 1
NjmCREQ	2	Same as above, but only defined if j>ICHG
NjmTREQ (j<NYRS)	all	REQ2(INDEX), where INDEX = ((k-1) x (NYRS+1)) + 1
WØmUBSG	4,5,6	UPBND(m)
TOTAUTH	all	AUTHMX(JGRADE)
URmn (m<n)	3,4,5,6	Ø.Ø25 x (NØmTREQ b-coefficient + NØnTREQ b-coefficient); i.e., 5 percent of average requirement for specialties m and n at $T_Ø$

[a] k is an integer number, in the range of 1-5Ø, indicating the position of the OPMS specialty number within the internal RHS file.

(16) Program: RANGE. - This program computes ranges on selected b-coefficients for the FMPS RANGE chapter input for all grades except Grade 2. Constraint names, grades, and coefficients are as shown in Table XI-18.

TABLE XI-18, Ranges on Selected b-coefficients Computed by the RANGE Program for FMPS RANGE Chapter Input

Constraint names	Grade	RANGE coefficients
NjmTREQ	3	REQ2(INDEX) x (OFLOHI) x (j+2), where INDEX = (m-1) x NSPEC x (NYRS+1) + j+1; computed only if m is an AES
URmn (m<n)	3,4,5,6	Ø.Ø25 x (NØmTREQ b-coefficient + NØnTREQ b-coefficient); i.e., 5 percent of average requirement for specialties m and n at $T_{Ø}$

(17) Program: BOUNDS. - This program defines bounds (upper, lower, fixed) for certain types of variables for the FMPS BOUNDS chapter input, as shown in Table XI-19.

TABLE XI-19, Upper, Lower, and Fixed Bounds Defined by the BOUNDS Program for FMPS BOUNDS Chapter Input

Variable	Grade	Bound type	Bound value
XØØØn (where n = selected specialties identified for control of input)	4,5,6	UP	UPBND(n) - REQ2(INDEX), where INDEX = (n-1) x (NYRS+1) + 1; (i.e., total budget authorization for Specialty n, minus the total requirements for Specialty n at $T_{Ø}$)
XØØØn	2	UP	REQ2(INDEX), where INDEX = (m-1) x (NYRS+1) + 1
XØmn (m=n for RYTPE code = 3)	4,5,6	FX	Ø.Ø (meaning that n is not preferred by any Specialty m)
XNn	3,4,5,6	UP	REQ2(INDEX), where INDEX = (m-1) x (NYRS+1) x (NSPEC+1)
XNn	3	LO	REQ2(INDEX) x (OFLOHI) x (NYRS+2); computed only if n is an AES

(18) Program: MASK. - This program creates the row and column selection masks used for the FMPS LOADLIST procedure. The masks are used to select solution records to write to the ODSOLUD1 file (which is input to the LINKAGE program). The row selection masks are for CREQ, TREQ, TOTAUTH, and UBSG records. The column selection masks are for XNn variables only.

(19) Program: SORTW. - This program sorts the records produced in the LOCOLS subroutine for W arcs (on file ODSAPUDØ4) in ascending sequence on the field composed of the year, specialty 1, and specialty 2 data values (e.g., for WØ1145 record, the "Ø1145" is the sort field).

(20) Program: SORTXY. - This program sorts the records produced in the LOCOLC, LOCOLS and RES2GR subroutines for X and Y arcs (on file ODSAPUDØ8) in ascending sequence on the field composed of the arc identification (e.g., XØ1145 or YØ1145).

b. Data Base Creation Activity

(1) Program: DATABASE. - This is the "executive" program in the data base creation activity. The program calls the subroutines NEWSAV, COMBIN, and RECORD, which create the necessary card images to be input to the data base load program (DBGEN) for both the current segment and cumulative data bases.

(2) Program: NEWSAV

(a) This program reads the ODDBSUD1 output file from the FMPS activity in groups of four 60 double-precision word records (the general format of these records is described in Chapter 2 of the FMPS programmers reference manual).

(b) The MASKCK subroutine (described in subparagraph (3) below) is called first to mask out unnecessary row data. The NEWSAV program then decodes the row or column information within each record to determine the identification and data values for each row/column, and writes a card image record out to file 1Ø, to be processed by the COMBIN subroutine. The following data values are included for each row or column name:

1. activity value

2. lower limit

3. upper limit

4. dual activity (for rows) or reduced cost (for columns)

(c) The NEWSAV program output record format is as shown in Table XI-20.

(3) Program: MASKCK

(a) This subroutine was added when a "bug" in level 6R1C of FMPS was detected. Selection masks for the rows to be written to ODDBSUD1 caused no row data to be written. The quick and easy solution was to bypass the row selection masks and thus write data for all rows to ODDBSUD1, and then have MASKCK screen out the unwanted rows.

(b) This subroutine checks all row records and bypasses the normal processing of the NEWSAV program for the following types of row records:

1. NjnGOZO

2. NjnLINC

3. NjnCINC

4. R_ _ _ _ _

5. RES_ _ _ _ _ (field grades only)

(4) Program: COMBIN

(a) This program combines the data generated in the matrix generator for W, X, and Y prefix column records; row records are read in and immediately written out to file ODSAPUDØ7. The following types of column records are also read in and immediately written out:

1. XØØØn

2. XNn

3. Xjmn (where m=n)

4. Yjmn (where m=n)

TABLE XI-20. Output Record Format for the NEWSAV Program

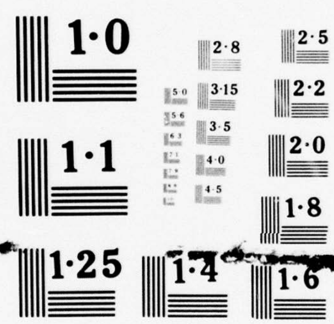
Print column	Data element	Values/description
1	Type	R, for row; C, for column
2	Prefix	V, N, T, U when Type is R X, Y, W when Type is C
3	Year	Ø-9 when Prefix is X, Y, V or N Ø when Prefix is W, T or U
4-5	Specialty m	Ø1-99 (actually OPMS specialty numbers only) when Prefix is X, Y, W, V, U, or N Ø when Prefix is T
6-7	Specialty n	Same as for Specialty m above
8	Grade	2-6 when Prefix is X, N or T 2-3 when Prefix is V 3-6 when Prefix is Y, U or W
9	Segment	1 when Prefix is V 1-2 when Prefix is X, W, U or T 1-4[a] when Prefix is N 3-4[b] when Prefix is Y
10-21	Activity	12 character FMPS solution value
22-23	Lower limit	12 character FMPS solution value
34-45	Upper limit	12 character FMPS solution value

[a]The N__CREQ records are assigned a Segment number 3 or 4 and a grade number one higher than the current segment. Additionally, the solution values for activity and upper limit are subtracted from the corresponding values for the N___TREQ constraint (same year and specialty so that the TREQ constraint solution values represent only requirements for one grade. The N___CREQ records are later combined with the N___TREQ records with the same year, specialty and grade values in the COMBIN subroutine.

[b]A Y prefix column record is assigned a Segment number 3 or 4 and a grade number one higher than that of the current segment. These records are later combined with X records with the same year, specialties and grade values in the COMBIN subroutine.

AD-A040 832 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 5/9
OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS) DOCUMENTATION.(U)
MAY 77 J D THOMAS, G E ARMSTRONG
UNCLASSIFIED CAA-D-77-1 NL

4 OF 5
ADA
040832



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

(b) The remaining types of column records, Xjmn and Yjmn (m≠n for both), are written out with the matrix generator information obtained from file ODSAPUDØ4 on W arc and X arc identification (and percentages thereof) which feed the arc identified by Xjmn or Yjmn. The output record format is as described in Table XI-21.

Table XI-21, Output Record Format for the COMBIN Program (continued on next page)

Print column	Data element	Values/description
1	Type	R, for row; C, for column
2	Prefix	N, T, U, V, W, X, Y
3	Year	Ø-9
4-5	Spec 1	Ø1-99 (actually OPMS specialties only)
6-7	Spec 2	Ø1-99 (actually OPMS specialties only)
8	Grade	2-6
9	Segment	1-4
10-21	Activity	12 character FMPS solution value
22-33	Lower Limit	12 character FMPS solution value
34-45	Upper Limit	12 character FMPS solution value
46	MRATIO	For either type R or type C: the left member of the utilization ratio for specialty pair defined by Spec 1/Spec 2
47	NRATIO	For type R: same as for MRATIO, above For type C: the right member of the utilization ratio for specialty pair defined by Spec 1/Spec 2

TABLE XI-21, Output Record Format for the COMBIN Program (concluded)

Print column	Data element	Values/description
48	Tour length of specialty m	Years
49	Tour length of specialty n	Years
50-54	Identification of X source	5-character ID
55-58	Percentage of X source	A percentage value converted to a 4-digit integer by multiplying by 1Ø,ØØØ (e.g., 33%, originally expressed as .33, would be printed out as 33ØØ)
59-63	Identification of W source	5-character ID
64-67	Percentage of W source	Derived in the same manner as Percentage of X source, above
68-72	Identification of alternate X source	5-character ID
73-76	Percentage of alternate X	Derived in the same manner as Percentage of X source, above

(5) Program: RECORD. - This program creates the input records for the cumulative data base by adding the solution values for appropriate X and Y arcs with the same year, specialties and grade identifiers, as well as adding the solution values of the N_ _ _CREQ and N_ _ _TREQ rows with corresponding year, specialty and grade identifiers. All other record types are unaffected by this program.

(a) The current segment data base input file (ODSAPUDØ7) is read, a record at a time, into array "A". If the record is not a row record (N_ _ _CREQ, N_ _ _TREQ, W_ _UBSG,

TOTAUTH, or RES_ _ _ _ _ type), or a column record for a W arc, then the first two characters of the record are changed to "ZZ". If it is a column record for a W arc, then the first two characters are changed to "WW". All the records are then fed to the SSREL sort subroutine.

(b) The permanent cumulative data base input file (ODSAPUD18) is also read, a record at a time, into array "A" and similar tests and modifications performed. The records are also fed to the SSREL sort subroutine.

(c) The records from both files are then combined and sorted. Records with "WW" are then changed back to "CW", and those with "ZZ" changed back to "CZ". The records with the same year, specialties, and grade are combined to form a single record, with the activity, lower limit and upper limit fields being the sum of the corresponding fields in the two input records. An exception to this procedure applies for column records; in these instances, the lower bound field is used to hold the activity value for the Y arc component of the Prefix "Z" record (Z record is composed of X and Y arc solution values along with identifying information).

(d) The combined records are then written to a temporary cumulative data base input file (ODSAPUDØ3) in card image format as shown in Table XI-22.

TABLE XI-22. Format of Records Written to Temporary Cumulative Data Base Input File ODSAPUDØ3 by the RECORD Program

Input column	Data element	Values/description
1	Type	R, for row; C, for column
2	Prefix	N, T, U, V for Type R W, Z for Type C
3	Year	Ø-9 when Prefix is N, V or Z; Ø when Prefix is T, U or W
4-5	Spec 1	Ø1-99 (actually OPMS specialties only) when Prefix is N, U, V, W or Z Ø when Prefix is T
6-7	Spec 2	Ø when Prefix is N, T or U; Ø1-99 (actually OPMS specialties only) when Prefix is V, W or Z
8	Grade	Ø2-Ø6
9	Segment	Ø
10-21	Activity	12-character FMPS solution value
22-33	Lower limit	12-character FMPS solution value
34-45	Upper limit	12-character FMPS solution value
46	MRATIO (Utilization ratio of Specialty m)	For either Type R or Type C: the left member of the utilization ratio for specialty pair defined by Spec 1/Spec 2
47	NRATIO (Utilization ratio of Specialty n)	For Type R: same as for MRATIO, above For Type C: the right hand member of the utilization ratio for specialty pair defined by Spec 1/ Spec 2
48	Tour length of Specialty m	Years
49	Tour length of Specialty n	Years

(6) Program: DB-CORRECT. - This program updates the activity and upper limit fields of the XNn arcs.

(a) The RECORD subroutine, described in (4) above, adds the NjnCREQ activity and upper limit solution values for the current segment for specialty n in all years j (except the last year) of the time span being analyzed, to the corresponding solution values for the NjnTREQ constraints from the cumulative data base file (the NjnCREQ and NjnTREQ pertain to the same grade). The updating of the XNn arcs provides for the necessary correction for the last year in the time span.

(b) Since ODSAS employs logical upper bounds on a variable for the total requirements for a specialty in the last year of the time span and a constraint for the unfilled higher grade requirements (same specialty and year), the corresponding column and row data for these node capacities must be extracted from the solution values for the appropriate row and column records. To accomplish this adjustment, the temporary cumulative data base file, ODSAPUDØ3, is first split into two temporary files (files 2Ø and 21) via the ED processor. File 2Ø contains all the records for constraints (rows), and file 21 contains all the records for the variables (columns).

(c) The DB-CORRECT program then reads files 2Ø and 21, saves the activity values for the NjnCREQ rows, and adds these values to the activity value of the XNn column records (same specialty, year and grade). A check is also made to ensure that the combined activity value does not exceed the original upper limit.

(d) During the processing, all the row and column records are written to a temporary file (file 22). At the completion of all updating, file 22 is copied to ODSAPUDØ3--except in the processing phase of the COL segment. In the latter case, since no NjnCREQ records are produced, no updating is required and thus the cumulative data base file (ODSAPUDØ3) is left alone.

(7) Program: DBGEN. - This program reads the data base input file (ODSAPUDØ3, ODSAPUD18, ODSAPUDØ7 or ODCUMUDØ1) and stores the data in the MIRADS-format data base file, MASfn, where fn is either CUMSEG (for cumulative data base file) or CURSEG (for current segment data base file). An accounting of the number of input records is taken and printed for user verification of the size of the data base.

(8) Program: HITFILE/INTERFACE. - This program reads the indices of the records selected via the MIRADS query set (named PROCUREMENT) from the file named HITFILE. The indices are stored in

the first word of 3-word records, 140 records to a block. MIRADS subroutines (READS) are used to read the HITFILE, extract the indices, and then select from the data base file (MASCUMSEG) those records at the index location. The selected records (which are in the format described in the data base dictionary) are written out to logical unit 9, which can then be accessed by user-written programs.

(9) Program: PROCUREMENT. - This program reads the file of data base records (ODSAPUD1Ø) selected from the CUMSEG data base via the "PROCUREMENT" query set. ODSAPUD1Ø contains the specialty pairs from the MAJ and CPT segment FMPS solutions and requirements data from the LT segment solution. The program has three major functions:

(a) The first function is to compute the unfilled requirements for each specialty by first computing the number promoted at $T_{\emptyset}$ and the attrition of promotees and non-promotees from $T_{\emptyset}$ to T_1, then finding the difference between the specialty's (node) upper limit and the activity and adding that difference to the total attrition.

(b) In the second function, PROCUREMENT computes up to three percentages for each specialty depending on whether the specialty is a BES or an AES and if the pair of specialties is a combination of either BES/BES, AES/BES or AES/AES. For all specialty pairs (i/j), the percent of Specialty i's requirements satisfied by the activity of the i/j pair is computed. For AES/BES specialty pairs (i/j), if the BES is not a combat arm, two additional percentages are computed--one is the percentage of Specialty i's requirements (excluding those satisfied by an AES/AES specialty pair) satisfied by the activity of the i/j pair, and the second percentage is of Specialty i's requirements when the activity of those pairs where the BES is a combat arm is also excluded. For AES/BES specialty pairs (i/j), where the BES is a combat arm, only the former percentage is computed.

(c) The third function produces the ODSAS Procurement Report in two parts. Part 1 details the procurement requirements for each BES, to include the AES requirements prorated to each BES based upon one of the percentages computed above (i.e., the percentage of Specialty i's requirements when only the AES/AES pairs are excluded). Part 2 is similar to Part 1, except that the percentage used is one wherein the AES/AES pairs and the pairs with combat arms are excluded. A third part (OPTION 1), a modified version of Part 2, may be produced at the user's option (see Chapter V).

c. <u>On-Line Inquiry Activity</u>. - This activity contains a single program--the UPDATE Program. UPDATE can be used to update (change) data values in either the master requirements file (ODSACUDØ1) or the attrition and promotion rates file (ODPOPUDØ1), or both. Parameter cards are required, to define which files are to be updated and which data type and values are to be changed. Input card requirements and formats are described fully in Paragraph 3a, Chapter IV.

d. <u>Segment Linkage Activity</u>. - This activity contains a single program--the LINKAGE Program--which has two major functions, described in Subparagraphs (1) and (2) below, respectively.

(1) The first function is to read the solution file (ODSOLUD1) created in the FMPS-MODULE and compute the unfilled requirements to be filled in the next grade segment.

(a) ODSOLUD1 is composed of records (each containing 6Ø double-precision words) for the node capacity constraints (NjnTREQ and NjnCREQ), and the XØØØn (for Grade 2 only) and XNn arcs. For each such constraint/arc the following information is included in the file:

<u>1</u>. Name of row/column

<u>2</u>. Activity code

<u>3</u>. Activity

<u>4</u>. Lower limit

<u>5</u>. Upper limit

(b) The internal subroutine, MODIFY, computes the unfilled requirements, depending upon segmentation options and the degree of satisfying the force structure requirements.

(c) The program then tests 12 sets of conditions and takes the updating actions shown in Table XI-23. Definitions of the terms used in Table XI-23 are as follows:

Term	Definition
Segmentation option	Ø - no segmentation within grade 1 - Segment 1 within a grade 2 - Segment 2 within a grade

Term	Definition
Constraint/Variable type	TREQ - network capacity constraint for total requirements CREQ - network capacity constraint for portion of total requirements pertaining to unfilled higher grades
ACTIVY	Solution value for variable or b-coefficient
CRQACT	ACTIVY value for a CREQ constraint (only used in Segment 1, since the ACTIVY value for CREQ is included in the ACTIVY value for TREQ; thus CRQACT is added back to determine proper adjustment when operating on a TREQ record)
ITREQ(YR,GRD,SP)	Internal requirements array indexed by year, grade and specialty
OFLOLO	Overflow percentage, which modifies requirements by allowing more than actual requirements for the grade being processed
REQONE	Requirements for next higher grade (unadjusted)
REQTOT	Sum of REQONE and REQTWO
REQTWO	Requirements for next higher grade (unadjusted)
TOTREQ	ULIMIT minus ACTIVY (total requirements less the solution value for the variable or b-coefficient)
ULIMIT	Upper limit specified on input
X1	Attrition = (ACTIVY x attrition rate)

TABLE XI-23, Updating Actions Taken by the LINKAGE Program

Segmentation option	Year	Constraint/ Variable type	ACTIVY relationship	Updating action
Ø or 2	Ø	TREQ	ACTIVY≤ULIMIT	ITREQ(YR,GRD,SP) = (TOTREQ)/(1.Ø+OFLOLO)
Ø or 2	1 to (NYRS-1)	TREQ	ACTIVY≤ULIMIT	ITREQ(YR,GRD,SP) = MAX(1.Ø, (TOTREQ+X1)/ 1.Ø+OFLOLO)
1	1 to NYRS	CREQ	ACTIVY<ULIMIT	ITREQ(YR,GRD+1,SP) = MAX(1.Ø,(REQONE-ACTIVY+X1)) then: save CRQACT = ACTIVY
1	1 to NYRS	CREQ	ACTIVY=ULIMIT	ITREQ(YR,GRD+1,SP) = MAX(1.Ø,(ITREQ(YR,GRD+1,SP) -ULIMIT+X1)) then: save CRQACT = ACTIVY
Ø, 1, or 2	N	XN_ _	ACTIVY≤ULIMIT	ITREQ(YR,GRD,SP) = (TOTREQ)/(1.Ø+OFLOLO)
Ø	Ø	XØØØn (grade 2 only)	ACTIVY≤ULIMIT	ITREQ(YR,GRD,SP) = (TOTREQ)/(1.Ø+OFLOLO)
1	Ø	TREQ	ACTIVY=ULIMIT	ITREQ(YR,GRD,SP) = MAX(Ø,(ITREQ(YR,GRD,SP)-ULIMIT))
1	Ø	TREQ	ACTIVY<ULIMIT	ITREQ(YR,GRD,SP) = MAX(Ø,(REQTWO+CRQACT-ACTIVY))
1	1-(NYRS-1)	TREQ	ACTIVY=ULIMIT	ITREQ(YR,GRD,SP) = MAX(1.Ø,(ITREQ(YR,GRD,SP) -ULIMIT+X1))
1	1-(NYRS-1)	TREQ	ACTIVY<ULIMIT	ITREQ(YR,GRD,SP) = MAX(1.Ø,(REQTWO+CRQACT -ACTIVY+X1))

(2) The second function of the LINKAGE program is to create the input parameters file (ODINPUDØ1) for the next system segment.

(a) In the first step in the process the last number on the first card image is changed to the grade number to be run. Then, the 4th through the 7th card images are replaced with the attrition/promotion rates from the ODPOPUDØ1 file for the appropriate grade (card images 2 and 3 are unchanged).

(b) The segmentation instructions for the grade segment just completed are then deleted and the segmentation instructions for the remaining field grade segments (if any) are moved up. When creating the input parameter file for the CPT and LT segments, the eighth card image contains 2Ø AES numbers, followed by the value of the variable ICHG and the number of AES. A ninth card image contains the rates used for alternate specialty designation in each segment.

(c) During processing, the new input parameter file is written to a temporary file (file 11.); at completion of the program execution, file 11. is copied to ODINPUDØ1.

e. Separating Active and Inactive Records in the Cumulative Data Base. - This is a special function which serves as an adjunct of the data base creation activity. The function is contained in the SEPARATE program, which separates the records that can no longer be updated from the remaining records in the current cumulative data base file, ODSAPUD18.

(1) The grade identification is read from the element SEGMENT in TPF$ (which was previously loaded in the catalogued runstream). File ODSAPUD18 is then read and, based on the initially read grade identification, any record whose grade field data value exceeds the grade identification plus 1, is written to temporary file 17; all others are written to temporary file 16.

(2) At the conclusion of the program, file 17 is placed at the the end of ODCUMUDØ1, and file 16 is copied into ODSAPUD18. For example, if the grade identification were "4", then Grade 6 records would be placed on ODCUMUDØ1, since the processing phases for MAJ and below would not produce records that could update any Grade 6 records.

(3) The SEPARATE program is portrayed graphically in Chapter III (Figure III-23).

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

CHAPTER XII
PROGRAM LISTINGS

1. Purpose. - The purpose of this chapter is to provide listings of all the source programs written for ODSAS. Since the FMPS and MIRADS are proprietary programs, their source listings are not available. The source program listings are arranged in alphabetical order, by activity within phase (initialization and processing). Each program listing has pertinent program documentation in a header block at the beginning of the listing and comment cards throughout the listing.

2. Index of Source Program Listings

a. Initialization Phase

Activity	Program	Page
(compute attrition/ promotion rates and create input parameters file)	INITIAL	XII-3
(create positional requirements file)	SACSCREATE	XII-17
	SACSEXTRACT	XII-23
	SACSPREPRO	XII-25
(create preference files)	TOURATIOS	XII-28

b. Processing Phase

Activity	Program	Page
Matrix Generator	BLOCK DATA	XII-32
	BOUNDS	XII-33
	CPDIAG	XII-35
	HICOL	XII-39
	HICOLS	XII-40
	IBITS	XII-45
	INPUT	XII-46
	IPHASE	XII-51
	IPROB	XII-52

Activity	Program	Page
Matrix Generator (cont)	JPHASE	XII-53
	KEYARC	XII-54
	LOCOL	XII-55
	LOCOLC	XII-57
	LOCOLL	XII-61
	LOCOLS	XII-63
	LODIAG	XII-68
	MAIN	XII-73
	MASK	XII-74
	OPT	XII-75
	PROC	XII-76
	RANGE	XII-77
	RESHI	XII-79
	RESLO	XII-85
	RES2GR	XII-88
	RHS	XII-92
	ROWCHP	XII-95
	ROWOP	XII-97
	SORTW	XII-99
	SORTXY	XII-100
	VALID	XII-101
Data Base Creation	COMBIN	XII-102
	DATABASE	XII-104
	DB-CORRECT	XII-105
	DBGEN	XII-106
	HITFILE/INTERFACE	XII-107
	MASKCK	XII-108
	NEWSAV	XII-109
	PROCUREMENT	XII-112
	RECORD	XII-118
On-Line Inquiry	UPDATE	XII-120
Segment Linkage	KTREQ	XII-123
	LINKAGE	XII-124
Separating Active and Inactive Records in the Cumulative Data Base	SEPARATE	XII-132

```
C     ******************************************************************
C     *                                                                *
C     *          PROGRAM: PFODSAP.INITIAL                              *
C     *          CALLING ARGUMENTS: NONE                               *
C     *          CALLED ROUTINES: READR8(INTERNAL)                     *
C     *                           VALID(INTERNAL)                      *
C     *          INPUT FILES: ODR8SUD01                                *
C     *          OUTPUT FILES: ODPOPUD01.                              *
C     *                        ODINPUD02  .                            *
C     *          PURPOSE: THIS PROGRAM COMPUTES ATTRITION ,PROMOTION,  *
C     *          AND OVER AND UNDER FLOW RATES FOR GRADES O-6 THRU O-2 *
C     *          FOR INPUT TO THE MATRIX GENERATOR PROGRAM. IT ALSO SETS *
C     *           UP THE INPUT FILE FOR FIRST SEGMENT.                 *
C     *          DATE: 15 APRIL 76                                     *
C     *          AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *          COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                   8120 WOODMONT AVENUE                         *
C     *                   BETHESDA,MARYLAND 20014                      *
C     *                                                                *
C     ******************************************************************
      COMMON ISPEC
      DIMENSION POP(6,32),R8WPR(6,32),RATE(7,31),ATTHI(6,9),ATTLO(6,9),
     1PRMT(6,9),AUTHMX(6),R8WOPR(6,32),WORK1(32),CPT(15)
     2,WORK2(32),CPTREM(15),VAL(32),REMMAJ(32),PROMAJ(32),ATRMAJ(32),
     3PMTCPT(15),PMTLT(15),ATRCPT(15),POPULA(1,4),IRATE(7,31),WORK3(32)
     4,SARRAY(3,50),UPBND(3,50),ISPEC(50),VAL1(32)
      INTEGER SPECLT(50),YEAR,GRADE,BGNYOS,ENDYOS,AES(50)
      INTEGER SEGCOL,SEGLTC,SEGMAJ
      EQUIVALENCE (RATE(1,1),IRATE(1,1))
      REAL LTR8W(2,30),LTR8WO(2,30)
       DATA SARRAY/150 * 1.0/
      DATA ISPEC/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
     145,46,47,48,49,51,52,53,54,71,72,73,74,75,76,77,81,82,83,86,87,88,
     291,92,93,95,97,00,00,00,00,00/
C
C                         *****FORMAT STATEMENTS*****
C
10    FORMAT (I2,2X,I2,3X,2A4,2X,A4,I1)
20    FORMAT (I
30    FORMAT (2I2,2A4)
40    FORMAT (8(F6.4,','),F6.4)
50    FORMAT (3(F6.4,','),F6.4)
60    FORMAT (5(F7.0,','),F7.0)
70    FORMAT (9(I2,','),I2)
80    FORMAT (I4,1X,'IS NOT A VALID NUMBER OF SPECIALTIES')
90    FORMAT (I4,1X,'IS NOT A VALID NUMBER OF YEARS')
100   FORMAT (1H0,'POPULATION SUM',7X,F12.1,10X,'ATTRITION   ',F13.2)
105   FORMAT (1H0,'ATTRITION RATE IN GRADE 6 = ',F13.2,' / ',F12.1,
     1' = ',F12.4)
110   FORMAT (1H0,11X,I2,'-',I2,3X,F12.1,5X,F7.4,9X,F7.4,12X,F13.2,12X,
     1F9.2,5X,F9.2)
120   FORMAT (1H0,11X,I2,'-',I2,5X,F12.1,7X,F7.4,7X,F13.2,18X,F7.4,7X,
     1F13.2)
130   FORMAT (1H1,20X,'****OFFICERS GRADE-',I1,' AT T-',I1,'****'/60X,
     1'ATTRITION'/1H ,12X,'YOS',7X ,'POPULATION',4X,'RATE W/PROM' 5X,
```

```
      2'(POPULATION*RATE W/PROM)')
135   FORMAT (1H0,F10.2,' -',F10.2,' =',F10.2)
140   FORMAT (1H1,30X,'****ATTRITION RATES FOR GRADE O-',I1,
     1' UPON PROMOTION TO O-',I1,//1H0,12X,'YOS',4X,'O-',I1,1X,
     2'PROMOTED BEFORE T-',I1,4X,'NEW-PROMOTEES',4X,'TOTAL PROM',4X,
     3'ATTR. RATE FOR GRADE-',I1,4X,'ATTRITION')
150   FORMAT (1H1,50X,'****OFFICERS GRADE-',I1,' AT T-',I1,'****'/1H0,
     122X,'STARTING',44X,'PROMOTIONS',12X,'POPULATION',5X,'ATTRITION',/
     213X,'YOS',6X,'POPULATION',3X,'RATE W/O PROM',4X,'RATE W/PROM',3X,
     3'(POPULATION*RATE DIFFERENCE)',3X,9X,6X,'IN GRADE')
160   FORMAT (1H0,'TOTAL POPULATION',3X,F12.1,24X,'TOTAL PROMOTIONS',
     1F13.2,3X,'TOTALS',3X,F9.2,5X,F9.2/1H0,13X,'ATTRITION RATE IN GRADE
     2 ',F9.2,'/',F9.2,' = ',F7.4/1H0,13X,'PROMOTION RATE ',9X,F13.2,
     3'/',F13.2,' = ',F7.4)
170   FORMAT (1H0,11X,I2,'-',I2,9X,F12.1,12X,F9.2,6X,F9.2,12X,F7.4,11X,
     1F9.2)
180   FORMAT (1H0,12X,'TOTALS',5X,F12.1,12X,F9.2,6X,F9.2,30X,F9.2)
190   FORMAT (1H0,'ATTRITION RATE',16X,'(',F10.2,' +',F10.2,')','/','(',
     1F10.2,' +',F10.2,')',' =',F10.2,'/',F10.2,' =',F7.4,/1X,
     2'FOR CPT AND MAJ WITH TWO SPECIALTIES')
200   FORMAT (1H0,'NO. OF AES EXCEEDED 20----FIRST 20 AES ACCEPTED'/)
210   FORMAT (1H0,10X,'SUBTOTALS  ',F12.1,21X,F13.2)
220   FORMAT (1H0,'CAPTAINS REMAINING T-',I1,' THRU T-',I1,' =',F12.1,
     1'/',F12.1,' =',F12.5)
230   FORMAT (1H0,'ATTRITION RATE IN GRADE 3 FOR THOSE WITH ONE SPECIALT
     1Y =',F12.1,' /',F12.1,' =',F12.5)
240   FORMAT (1H0,3X,'TOTAL POPULATION  ',F12.1,4X,'TOTAL ATTRITION',2X,
     1F13.2,15X,'TOTAL ATTRITION',2X,F13.2)
250   FORMAT (1H1,30X,'***CAPTAINS WITH TWO SPECIALTIES AT T-',I1,'***'/
     1/37X,'RATE W/PROM',13X,'ATTRITION',11X,'RATE W/O PROM'12X,
     2'ATTRITION'/1H0,12X,'YOS',7X,'POP/2 SPEC',5X,'(GRADE 3)',6X,
     3'(POP/2 SPEC*RATE W/PROM)',6X,'(GRADE 3)',6X,'(POP/2 SPEC*RATE W/O
     4 PROM)')
260   FORMAT (1H0,30X,'****CAPTAINS PROMOTED TO MAJOR AT T-',I1,'****'//
     168X,'RATE W/PROM',12X,'ATTRITION'/1H ,12X,'YOS',10X,'PROM TO MAJ',
     26X,'MAJ REMNG',5X,'TOT. MAJ',6X,'(GRADE 4)',5X,'(POPULATION*RATE W
     3/PROM)')
270   FORMAT (1H0,11X,I2,'-',I2,6X,F10.2,6X,F10.2,6X,F10.2,3X,F10.2,6X,
     1F10.2)
280   FORMAT (1H0,10X,'TOTALS',6X,F10.2,6X,F10.2,6X,F10.2,19X,F10.2)
290   FORMAT (1H1,50X,'****OFFICERS GRADE-',I1,' AT T-',I1,'****'/1H0,61
     1X,'ATTRITION'/1H ,103X,'PROMOTIONS',/13X,'YOS',9X,'POPULATION',4X,
     2'RATE W/PROM',4X,'(POPULATION*RATE W/PROM)',4X,'RATE W/O PROM',4X,
     3'(POPULATION*RATE DIFFERENCE)')
300   FORMAT (1H0,11X,I2,'-',I2,5X,F12.1,6X,F7.4,13X,F9.2,13X,F7.4,15X,
     1F12.1)
310   FORMAT (1H0,10X,'TOTAL POP  ',F12.1,8X,'TOTAL ATTR',8X,F9.2,14X,
     1'TOTAL PROM',11X,F12.1)
320   FORMAT(1H0,'***NUMBER OF ADVANCED ENTRY SPECIALTIES ',I3,
     *' EXCEED FILE SPACE - FIRST 20 WERE ACCEPTED')
325    FORMAT(1H0,'***NUMBER OF ADVANCED ENTRY SPECIALTIES INPUT ',
     *'DOES NOT AGREE WITH VARIABLE NBRAES')
330   FORMAT (1H0,'ATTRITION RATE IN GRADE',1X,I1,' =',F9.2,1X,'/',F9.2,
     11X,'=',F12.4)
340   FORMAT (1H1,53X,'****LIEUTENANTS AT T-',I1,'****'/112X'ATTRITION'/
     11H ,5X,'YOS',7X,'POPULATIONS',11X,'RATE W/PROM',8X,'RATE W/O PROM'
     2,6X,'PROMOTIONS',5X,'PROMOTIONS'1X,'((POP - PRMT)*RATE W/O PROM)'
```

```
     3/11X,'**********************',3X,'******************',3X,'**********
     4********',8X,'TO',14X,'TO',4X,'*****************************'/1H0,13X,
     5'1LT',9X,'2LT',8X,'1LT',7X,'2LT',7X,'1LT',7X,'2LT',10X,'CPT',12X,'
     61LT',8X,'1LT',6X,'2LT',7X,'TOT')
350   FORMAT (1H0,3X,I2,'-',I2,2X,F9.2,3X,F9.2,3X,F7.4,3X,F7.4,3X,F7.4,
     13X,F7.4,4X,F7.2,8X,F7.2,4X,F7.2,2X,F7.2,3X,F7.2)
360   FORMAT (1H0,2X,'TOTALS',2X,F9.2,3X,F9.2,43X,F8.2,7X,F8.2,3X,F8.2,
     12X,F7.2,3X,F7.2)
370   FORMAT (1H0,12X,'ATTRITION RATE FOR LIEUTENANTS',1X,F9.2,1X,'/',
     11X,'(',F9.2,1X,'+',F9.2,' - ',F8.2,' )',' = ',F8.4/1H0,12X,
     2'PROMOTION RATE FOR LIEUTENANTS'1X,F9.2,1X,'/',1X,'(',F9.2,1X,'+',
     3F9.2,1X,')',1X,'=',1X,F8.4)
380   FORMAT (1H0,43X,'****CAPTAINS POPULATION AT T-',I1,'****'/1H0,73X,
     1'RATE W/O PROM',14X,'ATTRITION'/1H ,5X,'YOS'7X,'PROM TO CPT',
     211X,'CPT REMAINING',9X,'TOT. CPT',8X,'(GRADE 3)',8X,'(TOT. CPT
     3*RATE W/O PROM)')
390   FORMAT (1H0,3X,I2,'-',I2,10X,F7.2,16X,F7.2,13X,F7.2,9X,F6.4,17X,
     1F7.2)
400   FORMAT (1H0,2X,'TOTALS',10X,F7.2,16X,F7.2,13X,F7.2,16X,
     1'TOTAL ATTRITION',1X,F7.2)
410   FORMAT (1H0,'CAPTAINS REMAINING =',F7.2,1X,'/',F7.2,1X,'=',F8.4)
420   FORMAT (1H0,'ATTRITION RATE IN GRADE 3 = ',F7.2,1X,'/',F7.2,1X,
     1'=',F8.4)
430   FORMAT (////1H0,'****ALL CAPTAINS AT T-0 WILL HAVE BEEN DESIGNATED
     1  ALTERNATE SPECIALTIES BY T-',I1,'*****')
435     FORMAT(///1H0,'****LIEUTENANTS AT T-0 WILL ATTAIN 8 YOS',
     1 ' BEGINNING AT T-',I1)
440   FORMAT (1H0,'CAPTAINS REMAINING T-0 PRIME THRU T-0 =','(',F12.1,
     1' +',F12.1,' )',' /',F12.1,' =',F12.5)
450   FORMAT (10F6.3)
460   FORMAT (10F6.0)
C
C     READ IN MODE OF DATA ***TEST FOR DUMMY DATA ***PROD*** FOR REAL***
C     READ IN PARAMETERS
      IGRADE=6
      NGRADE=6
      MODE='PROD'
      READ (5,30) NSPEC,NYRS,NAME1,NAME2
      IF (NSPEC.GT.0.AND.NSPEC.LE.50) GO TO 470
      WRITE (6,80) NSPEC
      STOP
470   IF (NYRS.GT.0.AND.NYRS.LE.9) GO TO 480
      WRITE (6,90) NYRS
      STOP
C     READ IN MAX. NUMBER OF OFFICERS BY GRADE
480   READ (5,20 ) AUTHMX
C     READ IN NO. OF SEGMENT-1 SPECIALTIES
C     READ IN SEGMENT-1 SPECIALTIES
      READ (5,20 ) NPRO,(SPECLT(K),K=1,NPRO)
C
C     CHECK FOR INVALID SPECIALTY
C
      CALL VALID (SPECLT,NPRO)
C         READ NUMBER OF AND IDENTITY OF ADVANCED ENTRY SPECIALTIES
          READ(5,20) NBRAES,(AES(K),K=1,NBRAES)
          IF(NBRAES.GT.20) WRITE(6,320)NBRAES
          IF(AES(NBRAES).EQ.0)WRITE(6,325)
```

```
      CALL VALID(AES,NBRAES)
C     READ IN POPULATION AND RATE DATA FOR ALL GRADES
      DO 500 K=1,NGRADE
      READ (5,20 ) GRADE,BGNYOS,ENDYOS
      READ (5,20 ) (POP(GRADE,J),J=BGNYOS,ENDYOS)
      READ (5,20 ) (R8WPR(GRADE,J),J=BGNYOS,ENDYOS)
C     IF GRADE .NE. 6 THEN READ IN R8WOPR DATA
      IF (K.NE.1) READ (5,20 ) (R8WOPR(GRADE,J),J=BGNYOS,ENDYOS)
C
C     SAVE FOR FURTHER COMPUTATION
      POP(GRADE,31)=BGNYOS
      POP(GRADE,32)=ENDYOS
C
C     SAVE OVERFLOW AND UNDERFLOW VARIABLES
      READ (5,20 ) (RATE(GRADE,L),L=28,31)
      IF (GRADE.GT.2) GO TO 500
C     READ IN 9 ADDITIONAL ATTRITION RATES FOR LIEUTENANTS
      IBGN=ENDYOS+1
      IEND=ENDYOS+9
      READ (5,20 ) (LTR8W(GRADE,J),J=IBGN,IEND)
      READ (5,20 ) (LTR8WO(GRADE,J),J=IBGN,IEND)
      DO 490 LL=BGNYOS,ENDYOS
      LTR8W(GRADE,LL)=R8WPR(GRADE,LL)
      LTR8WO(GRADE,LL)=R8WOPR(GRADE,LL)
490   CONTINUE
500   CONTINUE
        ...READ IN SEGMENTATION INSTRUCTIONS AND VALUES...
C
      READ (5,20 ) KGRADE,SEGCOL,LGRADE,SEGLTC,MGRADE,SEGMAJ
       CALL READR8 (6,SARRAY,UPBND,NPRO,SPECLT,SEGCOL,NSPEC)
       CALL READR8 (5,SARRAY,UPBND,NPRO,SPECLT,SEGLTC,NSPEC)
       CALL READR8 (4,SARRAY,UPBND,NPRO,SPECLT,SEGMAJ,NSPEC)
C     COMPUTE AND SAVE ATTHI,ATTLO,AND PRMT RATES
C
C     GRADE 6 COMPUTATION ONLY
C
      J=6
      L=0
      BGNYOS=POP(6,31)
      ENDYOS=POP(6,32)
      ICNT=ENDYOS
C     BEGIN LOOP FOR YEARLY RATES
      DO 530 YEAR=1,9
      WRITE (6,130) J,L
C     COMPUTE AND WRITE DETAIL LINES
      DO 510 JJ=BGNYOS,ENDYOS
      R8WOPR(6,JJ)=R8WPR(6,JJ)
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
      VALUE=POP(6,JJ)*R8WPR(6,LL)
      IF (NN.GT.ENDYOS) GO TO 510
      WRITE (6,120) NM,NN,POP(J,JJ),R8WPR(J,LL),VALUE
510   CONTINUE
C
      DO 520 K=BGNYOS,ENDYOS
      INDEX=K+L
```

```
      IF (INDEX.GT.ENDYOS) INDEX=1
C     SUM POPULATION AND ATTRITION
      POPULA(1,1)=POPULA(1,1)+POP(6,K)
      POPULA(1,2)=POPULA(1,2)+R8WPR(6,INDEX)*POP(6,K)
C     ADJUST POPULATIONS FOR ATTRITION
      POP(6,K)=POP(6,K)-R8WPR(6,INDEX)*POP(6,K)
      IF (K.NE.ICNT) GO TO 520
C     ADD LAST 2 YEARS TOGETHER TO GET 30 AND OVER CATEGORY
      KK=K-1
      POP(6,KK)=POP(6,KK)+POP(6,K)
      POP(6,K)=0.0
      ICNT=ICNT-1
520   CONTINUE
C     WRITE SUMMARY LINE OF ATTRITION RATE
      ATTLO(6,YEAR)=POPULA(1,2)/POPULA(1,1)
      WRITE (6,100) POPULA(1,1),POPULA(1,2)
      WRITE (6,105) POPULA(1,2),POPULA(1,1),ATTLO(6,YEAR)
      ATTHI(6,YEAR)=ATTLO(6,YEAR)
      PRMT(6,YEAR)=0.0
C
C     ZERO OUT FOR COMPUTATION PURPOSES
      POPULA(1,1)=0.0
      POPULA(1,2)=0.0
      L=L+1
530   CONTINUE
C
C     RATE COMPUTATION FOR GRADES 5 THRU 4
C
      DO 640 J=5,4,-1
      IGR=J+1
      L=0
C
C     SAVE POPULATION ARRAY FOR COMPUTATION OF ATTHI AND PRMT
      DO 540 M=1,32
      WORK1(M)=POP(J,M)
540   CONTINUE
      BGNYOS=WORK1(31)
      ENDYOS=WORK1(32)
      ICNT=ENDYOS
      JCNT=ENDYOS
      DO 550 M=1,32
      POP(J,M)=0.0
550   CONTINUE
C
C     BEGIN LOOP FOR YEARLY RATES
C      .
      DO 620 YEAR=1,9
      WRITE (6,150) J,L
C     COMPUTE AND WRITE DETAIL LINES
      DO 560 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
C      ...POPULATION SUM
      POPULA(1,1)=POPULA(1,1)+WORK1(JJ)
C     ....PROMOTIONS BY YEAR OF SERVICE
```

```
      WORK2(JJ)=(R8WPR(J,LL)-R8WOPR(J,LL))*WORK1(JJ)
C     *****PROMOTION SUM
      POPULA(1,3)=POPULA(1,3)+WORK2(JJ)
C     *****POPULATION  SUM
      POPULA(1,4)=POPULA(1,4)+WORK1(JJ)
C     **ATTRITION IN GRADE
      VALUE=WORK1(JJ)*R8WOPR(J,LL)
      WORK3(JJ)=VALUE
C     ****ATTRITION SUM
      POPULA(1,2)=POPULA(1,2)+VALUE
      IF (NN.GT.ENDYOS) GO TO 560
      WRITE (6,110) NM,NN,WORK1(JJ),R8WOPR(J,LL),R8WPR(J,LL),WORK2(JJ),
     1WORK1(JJ),VALUE
560   CONTINUE
C
      DO 570 K=BGNYOS,ENDYOS
      INDEX=K+L
      IF (INDEX.GT.ENDYOS) INDEX=1
C     ADJUST POPULATIONS FOR ATTRITION AND PROMOTION
      WORK1(K)=WORK1(K) - WORK2(K) - WORK3(K)
      IF (K.LT.ICNT) GO TO 570
C     ADD LAST 2 YEARS TO GET ENDYOS AND OVER CATEGORY
      KK=K-1
      WORK1(KK)=WORK1(KK)+WORK1(K)
      WORK1(K)=0.0
      ICNT=ICNT-1
      GO TO 580
570   CONTINUE
580   ATTLO(J,YEAR)=POPULA(1,2)/POPULA(1,4)
      PRMT(J,YEAR)=POPULA(1,3)/POPULA(1,1)
C     WRITE ATTRITION AND PROMOTION RATES
      WRITE (6,160) POPULA(1,1),POPULA(1,3),POPULA(1,4),POPULA(1,2),
     1POPULA(1,2),POPULA(1,4),ATTLO(J,YEAR),POPULA(1,3),POPULA(1,1),
     2PRMT(J,YEAR)
C     ****ATTHI COMPUTATIONS FOR GRADES 5 AND 4
      WRITE (6,140) J,IGR,IGR,L,IGR
      TOTATR=0.0
      RESUM=0.0
      POPSUM=0.0
C     COMPUTE AND WRITE DETAIL LINES
      DO 600 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
      SUM=POP(J,JJ)+WORK2(JJ)
      VALUE=SUM*R8WOPR(IGR,LL)
      IF (NN.GT.ENDYOS) GO TO 590
      WRITE (6,170) NM,NN,POP(J,JJ),WORK2(JJ),SUM,R8WOPR(IGR,LL),VALUE
590   POPSUM=POPSUM+SUM
      RESUM=RESUM+POP(J,LL)
      TOTATR=TOTATR+VALUE
      POP(J,JJ)=SUM-VALUE
```

```
      POP(J,JJ)=0.0
      JCNT=JCNT-1
      GO TO 610
600   CONTINUE
610   ATTHI(J,YEAR)=TOTATR/POPSUM
      WRITE (6,180) RESUM,POPULA(1,3),POPSUM,TOTATR
      WRITE (6,330) IGR,TOTATR,POPSUM,ATTHI(J,YEAR)
C
C     ZERO OUT FOR COMPUTATION PURPOSES
      POPULA(1,1)=0.0
      POPULA(1,2)=0.0
      POPULA(1,3)=0.0
      POPULA(1,4)=0.0
      L=L+1
620   CONTINUE
C
C     ZERO OUT FOR COMPUTATION PURPOSES
C
      DO 630 M=1,32
      WORK1(M)=0.0
      WORK2(M)=0.0
630   CONTINUE
640   CONTINUE
C
C     GRADE 3  COMPUTATION ONLY
C
      J=3
      IGR=J+1
      L=0
      TOTAL=0.0
C
C     CHECK AND SAVE THE NO. OF ADVANCED ENTRY SPECIALTIES
      IF (NBRAES.GT.20) WRITE (6,200)
C
C     SAVE THE FIRST 20 AES'S
      DO 650 K=1,20
      IRATE(1,K)=AES(K)
650   CONTINUE
C
C     SAVE POPULATION ARRAY FOR COMPUTATION OF ATTHI
C
      DO 660 M=1,32
      WORK1(M)=POP(J,M)
660   CONTINUE
      BGNYOS=WORK1(31)
      ENDYOS=WORK1(32)
      ICNT=ENDYOS
      JCNT=ENDYOS
C     COMPUTE WHEN LAST CAPTAINS WOULD EXCEED 8 YOS
      ICHG=9-BGNYOS
      WRITE (6,430) ICHG
      IRATE(1,31)=ICHG
      IRATE(1,30)=NBRAES
      IEND=8
C     BEGIN LOOP FOR YEARLY RATES
      DO 780 YEAR=1,9
      RSUBTL=0.0
```

```
      PSUBTL=0.0
      TOTPOP=0.0
      TOTATR=0.0
      TOTPRO=0.0
      POPMAJ=0.0
      WRITE (6,290) J,L
C     COMPUTE DETAIL LINES
      DO 670 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      IF (LL.GT.ENDYOS) LL=1
      VAL(LL)=WORK1(JJ)*R8WPR(J,LL)
      VAL1(LL)=WORK1(JJ)*R8WOPR(J,LL)
      PROMAJ(JJ)=WORK1(JJ)*(R8WPR(J,LL)-R8WOPR(J,LL))
      TOTPOP=TOTPOP+WORK1(JJ)
      TOTATR=TOTATR+VAL(LL)
      POPMAJ=POPMAJ+PROMAJ(JJ)
      IF (NN.GT.ENDYOS) GO TO 670
      IF (YEAR.GT.1) GO TO 670
      TOTAL=TOTAL+WORK1(JJ)
670   CONTINUE
      IF (IEND.LT.BGNYOS) GO TO 690
      DO 680 KK=BGNYOS,IEND
      LL=KK+L
      IF (LL.GT.ENDYOS) LL=1
      PSUBTL=PSUBTL+WORK1(KK)
      RSUBTL=RSUBTL+VAL(LL)
680   CONTINUE
C     COMPUTE CPTREM(1)
      IF (YEAR.GT.1) GO TO 690
      CPTREM(YEAR)=(PSUBTL+WORK1(9))/TOTPOP
      RATE(1,21)=CPTREM(YEAR)
C
C     WRITE DETAIL LINES
C
690   DO 710 K=BGNYOS,ENDYOS
      LL=K+L
      NN=K+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
      IF (NN.GT.ENDYOS) GO TO 710
      IF (LL.EQ.9) GO TO 700
      WRITE (6,300) NM,NN,WORK1(K),R8WPR(J,LL),VAL(LL),R8WOPR(J,LL),
     1PROMAJ(K)
      GO TO 710
700   WRITE (6,210) PSUBTL,RSUBTL
      WRITE (6,300) NM,NN,WORK1(K),R8WPR(J,LL),VAL(LL),R8WOPR(J,LL),
     1PROMAJ(K)
710   CONTINUE
      CPTREM(YEAR+1)=PSUBTL/TOTAL
      ATTLO(J,YEAR)=RSUBTL/PSUBTL
C     WRITE SUMMARY LINES AND RATES
      WRITE (6,310) TOTPOP,TOTATR,POPMAJ
      IF (L.EQ.0) WRITE (6,440) PSUBTL,WORK1(9),TOTPOP,CPTREM(YEAR)
      WRITE (6,220) L,YEAR,PSUBTL,TOTAL,CPTREM(YEAR+1)
      WRITE (6,230) RSUBTL,PSUBTL,ATTLO(J,YEAR)
C     SAVE CPTREM VALUE IN RATE ARRAY
```

```
      K=YEAR+21
      RATE(1,K)=CPTREM(YEAR+1)
      CPTPOP=0.0
      CPTATR=0.0
      CPTARN=0.0
      WRITE (6,250) L
C     COMPUTATIONS FOR CPTS W/2 SPECIALTIES AND DETAIL LINES
      DO 720 K=8,ENDYOS
      LL=K+L
      NN=K+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
      CPTPOP=CPTPOP+WORK1(K)
      CPTATR=CPTATR+VAL(LL)
      CPTARN=CPTARN+VAL1(LL)
      IF (NN.GT.ENDYOS) GO TO 720
      WRITE (6,120) NM,NN,WORK1(K),RBWPR(J,LL),VAL(LL),RBWOPR(J,LL),
     1VAL1(LL)
720   CONTINUE
      WRITE (6,240) CPTPOP,CPTATR,CPTARN
C     ***** ATTHI COMPUTATIONS FOR GRADE 3
      TATR4=0.0
      TOTREM=0.0
      TPROM=0.0
      POPMAJ=0.0
      WRITE (6,260) L
C     COMPUTE AND WRITE DETAIL LINES
      DO 730 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      IF (LL.GT.ENDYOS) LL=1
      IF (JJ .LT. 8) GO TO 725
      TPROM=TPROM+PROMAJ(JJ)
725   TOTPRO=TOTPRO+PROMAJ(JJ)
      TOTREM=TOTREM+REMMAJ(JJ)
      WORK2(JJ)=PROMAJ(JJ)+REMMAJ(JJ)
      ATRMAJ(JJ)=WORK2(JJ)*RBWPR(IGR,LL)
      TATR4=TATR4+ATRMAJ(JJ)
      POPMAJ=POPMAJ+WORK2(JJ)
      IF (NN.GT.ENDYOS) GO TO 730
      WRITE (6,270) NM,NN,PROMAJ(JJ),REMMAJ(JJ),WORK2(JJ),RBWPR(IGR,LL),
     1ATRMAJ(JJ)
730   CONTINUE
      PRODIF=POPMAJ-TPROM
      WRITE (6,280) TOTPRO,TOTREM,POPMAJ,TATR4
C     ADJUST MAJOR POPULATION FOR ATTRITION
      DO 740 K=BGNYOS,ENDYOS
      INDEX=K+L
      IF (INDEX.GT.ENDYOS) INDEX=1
      REMMAJ(K)=WORK2(K)-WORK2(K)*RBWPR(IGR,INDEX)
      WORK2(K)=WORK2(K)+REMMAJ(INDEX)+PROMAJ(INDEX)-ATRMAJ(INDEX)
      IF (K.LT.JCNT) GO TO 740
C     ADD LAST 2 YEARS TOGETHER TO GET ENDYOS AND OVER CATEGORY
      KK=K-1
      REMMAJ(KK)=REMMAJ(KK)+REMMAJ(K)
      REMMAJ(K)=0.0
```

```
      JCNT=JCNT-1
      GO TO 750
740   CONTINUE
750   TOTATR=CPTARN+TATR4
      TOTPOP=CPTPOP+PRODIF
      ATTHI(J,YEAR)=TOTATR/TOTPOP
C      WRITE SUMMARY LINES
      WRITE (6,190) TATR4,CPTARN,PRODIF,CPTPOP,TOTATR,TOTPOP,ATTHI(J,
     1YEAR)
      WRITE (6,135) POPMAJ,TPROM,PRODIF
      DO 760 K=BGNYOS,ENDYOS
      INDEX=K+L
      IF ((INDEX.GT.ENDYOS) INDEX=1
C     ADJUST POPULATIONS FOR ATTRITION
      WORK1(K)=WORK1(K)-RAWPR(J,INDEX)*WORK1(K)
      IF (K.LT.ICNT) GO TO 760
C     ADD LAST 2 YEARS TOGETHER TO GET ENDYOS AND OVER CATEGORY
      KK=K-1
      WORK1(KK)=WORK1(KK)+WORK1(K)
      WORK1(K)=0.0
      ICNT=ICNT-1
      GO TO 770
760   CONTINUE
770   IEND=IEND-1
      L=L+1
780   CONTINUE
C
C     RATE COMPUTATIONS FOR LIEUTENANTS
C
      L=0
C
C     SAVE POPULATION ARRAYS FOR COMPUTATION OF ATTHI
      DO 790 M=1,32
      WORK1(M)=POP(2,M)
      WORK2(M)=POP(1,M)
790   CONTINUE
      BGNYOS=WORK1(31)
      ENDYOS=WORK1(32)
      ICHG=9-ENDYOS
      IRATE(7,10)=ICHG
C     ZERO OUT CPT REMAINING ARRAY
      DO 800 J=1,15
      CPTREM(J)=0.0
800   CONTINUE
C     BEGIN LOOP FOR YEARLY RATES
C
      DO 860 YEAR=1,9
      POP1LT=0.0
      POP2LT=0.0
      TOTATR=0.0
      CPTATR=0.0
      TATR2=0.0
      TATR1=0.0
      TPRMT3=0.0
      TPRMT2=0.0
      WRITE (6,340) L
C     COMPUTE AND WRITE DETAIL LINES
```

```
      DO 810 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      PMTCPT(JJ)=WORK1(JJ)*(LTRBW(2,LL)-LTRBWO(2,LL))
      PMT1LT(JJ)=WORK2(JJ)*(LTRBW(1,LL)-LTRBWO(1,LL))
      VAL1LT=(WORK1(JJ)-PMTCPT(JJ))*LTRBWO(2,LL)
      VAL2LT=WORK2(JJ)*LTRBWO(1,LL)
      TATR1=TATR1+VAL2LT
      TATR2=TATR2+VAL1LT
      ATRTOT=VAL1LT+VAL2LT
      TOTATR=TOTATR+ATRTOT
      TPRMT3=TPRMT3+PMTCPT(JJ)
      TPRMT2=TPRMT2+PMT1LT(JJ)
      POP1LT=POP1LT+WORK1(JJ)
      POP2LT=POP2LT+WORK2(JJ)
      WRITE (6,350) NM,NN,WORK1(JJ),WORK2(JJ),LTRBW(2,LL),LTRBW(1,LL),
     1LTRBWO(2,LL),LTRBWO(1,LL),PMTCPT(JJ),PMT1LT(JJ),VAL1LT,VAL2LT,
     2ATRTOT
810   CONTINUE
C     WRITE SUMMARY LINES
      WRITE (6,360) POP1LT,POP2LT,TPRMT3,TPRMT2,TATR2,TATR1,TOTATR
      ATTLO(2,YEAR)=TOTATR/(POP1LT+POP2LT)
      PRMT(2,YEAR)=TPRMT3/(POP1LT+POP2LT)
      WRITE (6,370) TOTATR,POP1LT,POP2LT,TPRMT3,ATTLO(2,YEAR),TPRMT3,
     1POP1LT,POP2LT,PRMT(2,YEAR)
C
C     **** ATTHI COMPUTATIONS FOR LIEUTENANTS
C
      TOTAL=0.0
      TOTPRO=0.0
      TATR3=0.0
      TOTREM=0.0
      CPTPOP=0.0
      WRITE (6,380) L
C     COMPUTE AND WRITE DETAIL LINES
      DO 830 JJ=BGNYOS,ENDYOS
      LL=JJ+L
      NN=JJ+L
      NM=NN-1
      TOTPRO=TOTPRO+PMTCPT(JJ)
      TOTREM=TOTREM+CPTREM(JJ)
      CPT(JJ)=PMTCPT(JJ)+CPTREM(JJ)
      ATRCPT(JJ)=CPT(JJ)*RBWOPR(3,LL)
      TATR3=TATR3+ATRCPT(JJ)
      CPTPOP=CPTPOP+CPT(JJ)
      IF (NN.GT.8) GO TO 820
      TOTAL=TOTAL+CPT(JJ)
820   WRITE (6,390) NM,NN,PMTCPT(JJ),CPTREM(JJ),CPT(JJ),RBWOPR(3,
     1LL),ATRCPT(JJ)
830   CONTINUE
      WRITE (6,400) TOTPRO,TOTREM,CPTPOP,TATR3
C
C     ADJUST CAPTAIN POPULATION
      DO 840 K=BGNYOS,ENDYOS
      INDEX=K+L
      CPTREM(K)=CPT(K)*(1.-RBWOPR(8,INDEX))
```

```
      CPT(K)=CPT(K)+CPTREM(K)+PMTCPT(K)-ATRCPT(K)
840   CONTINUE
      CPTRM=TOTAL/CPTPOP
      ATTHI(2,YEAR)=TATR3/CPTPOP
C     SAVE CAPTAIN REMAINING VALUE IN RATE ARRAY
      RATE(7,YEAR)=CPTRM
      WRITE (6,410) TOTAL,CPTPOP,CPTRM
      WRITE (6,420) TATR3,CPTPOP,ATTHI(2,YEAR)
      DO 850 K=BGNYOS,ENDYOS
      INDEX=K+L
C
C     ADJUST POPULATIONS FOR ATTRITION
      WORK1(K)=WORK1(K)-LTRBW(2,INDEX)*WORK1(K)+(PMTILT(K)*(1.-LTRBW(2,
     1INDEX)))
      WORK2(K)=WORK2(K)-LTRBW(1,INDEX)*WORK2(K)
850   CONTINUE
      L=L+1
860   CONTINUE
C
C     SAVE ATTHI,ATTLO, AND PRMT RATES
C
      DO 880 M=2,6
      DO 870 K=1,9
      J=K+9
      J1=J+9
      RATE(M,K)=ATTHI(M,K)
      RATE(M,J)=ATTLO(M,K)
      RATE(M,J1)=PRMT(M,K)
870   CONTINUE
880   CONTINUE
C          VERIFICATION OF ICHG  FOR LTS
         DO 900 YEAR = 1,9
          IF(RATE(7,YEAR).EQ.1.0) GO TO 900
          ICHG=YEAR-1
        IRATE(7,10)=ICHG
          GO TO 910
900       CONTINUE
910        WRITE(6,435)ICHG
           IRATE(1,30)=NBRAES
C     SAVE RATE DATA TO MODIFY DIFFERENT SEGMENTS
      WRITE (10) RATE
      END FILE 10
C     CREATE INPUT DATA FILE (F1INPUTDATA.)
C     WRITE PARAMETERS
      WRITE (11,10) NSPEC,NYRS,NAME1,NAME2,MODE,IGRADE
C     WRITE MAX. NO. OF OFFICERS IN GRADE
      WRITE (11,60) AUTHMX
C     WRITE SEGMENT-1 SPECIALTIES
      WRITE (11,70) NPRO,(SPECLT(J),J=1,NPRO)
C     WRITE HIGH ATTRITION RATES ***ATTHI***
      WRITE (11,40) (RATE(6,K),K=1,9)
C     WRITE LOW ATTRITION RATES ***ATTLO***
      WRITE (11,40) (RATE(6,K),K=10,18)
C     WRITE PROMOTION RATES ***PRMT***
      WRITE (11,40) (RATE(6,K),K=19,27)
C     WRITE OVER AND UNDER FLOW VARIABLES
      WRITE (11,50) (RATE(6,K),K=28,31)
```

```
C        ***WRITE OUT SEGMENTATION INSTRUCTIONS FOR GRADES 4-6***
         WRITE (11,20 ) SEGCOL
         WRITE (11,460) (UPBND(3,K),K=1,50)
         WRITE (11,450) (SARRAY(3,K),K=1,50)
         WRITE (11,20 ) SEGLTC
         WRITE (11,460) (UPBND(2,K),K=1,50)
         WRITE (11,450) (SARRAY(2,K),K=1,50)
         WRITE (11,20 ) SEGMAJ
         WRITE (11,460) (UPBND(1,K),K=1,50)
         WRITE (11,450) (SARRAY(1,K),K=1,50)
         END FILE 11
C
         STOP FINISH
C
C        INVALID SPECIALTY SUBROUTINE FOR VERIFICATION OF SPECIALTY NO.
C
         SUBROUTINE VALID (ISPECT,L)
         DIMENSION ISPEC(50),ISPECT(50)
         COMMON ISPEC
C
C                               *****FORMAT STATEMENTS*****
C
980      FORMAT (1H0,I3,' IS NOT A VALID SPECIALTY NUMBER')
C
         DO 1000 K=1,L
         DO 990 J=1,50
         IF (ISPECT(K).NE.ISPEC(J)) GO TO 990
         GO TO 1000
990      CONTINUE
         WRITE (6,980) ISPECT(K)
1000     CONTINUE
         RETURN
C
C          ***INTERNAL SUBROUTINE FOR READING AND VERIFYING
C               SEGMENTATION PARAMETERS****
C
         SUBROUTINE READR8 (NBRGRD,SARRAY,UPBND,NPRO,SPECLT,SEGMNT,NSPEC)
         DIMENSION ISPEC(50),MSPEC(50),SARRAY(3,50),UPBND(3,50),TEMP(50)
         COMMON ISPEC
            INTEGER SPECLT(50)
           INTEGER SEGMNT
C
C                               *****FORMAT STATEMENTS*****
C
1060     FORMAT ()
1070     FORMAT (1HC,'INVALID SPECIALTY NUMBER ENTERED WHEN READING ',
        1'SEGMENTATION PARAMETERS FOR GRADE',I2,' SPECIALTY',I3)
C
         INDEX=NBRGRD-3
         READ (5,1060) (MSPEC(K),SARRAY(INDEX,K),K=1,NSPEC)
          IF(SEGMNT.NE.0) GO TO 1076
          READ(5,1060) (MSPEC(K),UPBND(INDEX,K),K=1,NSPEC)
          RETURN
1076     IEND=NSPEC
C           VALIDATE SPECIALTY NUMBER ENTERED
1080     DO 1090 M=1,IEND
         IF (MSPEC(M).EQ.ISPEC(M)) GO TO 1090
```

```
      WRITE (6,1070) NBRGRD,ISPEC(M)
      IERR=IERR+1
1090  CONTINUE
      IF (IERR.GT.0) RETURN 0
        READ (5,1060) (MSPEC(K),TEMP(K),K=1,NSPEC)
C         VALIDATE NUMBER OF UPPER BOUND VALUES ENTERED
C           NEED ONE VALUL FOR EACH SPECIALTY IN SEGMENT 1
        ICNT=0
        DO 1100 J=1,NSPEC
        IF (TEMP(J) .EQ.1.0) GO TO 1100
        DO 1110 K=1,NPRO
        IF(MSPEC(J).NE.SPECLT(K)) GO TO 1110
         UPBND(INDEX,K) = TEMP(J)
        GO TO 1100
1110    CONTINUL
        ICNT=ICNT+1
        WRITE(6,1075) MSPEC(J),NBRGRD
1075    FORMAT(1H0,'UPPER BOUND DATA ERRONEOUSLY ENTERED FOR ',
     *'SPECIALTY NUMBER ',I2,' GRADE',I3)
1100    CONTINUE
        IF(ICNT.GT.0)RETURN 0
      RETURN
      END
```

```
C       ..............................................................
C       .                                                            .
C       .     PROGRAM:      .SACSCREATE                              .
C       .     INPUT FILES: 10= ODOUTUDO1.                            .
C       .     OUTPUT FILES: 11= UDSAPUD1U.                           .
C       .     PURPOSE:                                               .
C       .     THIS PROGRAM DERIVES FROM THE SACS AND LISTS OUT THE NO.  .
C       .     OF OFFICERS IN GRADES O-6 THRU O-2 THAT ARE REQUIRED FOR  .
C       .     A 10-YEAR PERIOD IN THE PRIMARY SPECIALTIES.           .
C       .     DATE: 15 APRIL 76                                      .
C       .     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN .
C       .     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            .
C       .              8120 WOODMONT AVENUE                          .
C       .              BETHESDA,MARYLAND 20014                       .
C       .                                                            .
C       ..............................................................
C
        DIMENSION ITREQ(11,6,100),NSPEC(50),ITIME(10)
        DATA NSPEC/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
       145,46,47,48,49,51,52,53,54,70,71,72,73,74,75,76,77,81,82,83,86,87,
       288,91,92,93,95,97,00,00,00,00/
         INTEGER  TOTREQ
    10 FORMAT (1X,I1,I2,15X,I6,I6,9X,I5)
12       FORMAT(1HO,'SPECIALTY ID ERROR IN FOLLOWING RECORD')
15       FORMAT(1X,A1,A2,15X,A6,A6,9X,A5)
    20 FORMAT ('1'/,49X,'TOTAL REQUIREMENTS FOR SPECIALTY',I3,/)
    21 FORMAT ('1'/,49X,'TOTAL REQUIREMENTS FOR GRADE',I2,/)
    30 FORMAT ('/ YEAR  OF',5(9X,'GRADE'),/2X,'SERVICE',5(11X,'O-',I1)/)
    31 FORMAT ('/ SPECIALTY',10(5X,'YEAR'),/16X,'T=0',9(6X,'T=',I1)/)
    40 FORMAT (4X,'T',I1,3X,5I14,/)
    41 FORMAT (4X,I2,5X,10(3X,I6))
    50 FORMAT (9HO  TOTALS,5I14)
    51 FORMAT (9HO  TOTALS,2X,10(3X,I6))
    60 FORMAT ('1  END PROGRAM. RECORDS = ',I6,3X,'NUMBER OF 00 = ',I6//)
    70 FORMAT (I6)
75       FORMAT(1HO,'REDISTRIBUTION OF SPECIALTY 70 REQUIREMENTS',
       *' TO LOGISTICS SPECIALTIES')
    80 FORMAT (2X,'T-0 IS ',I6,/)
85       FORMAT(1HO,'TOTAL REQUIREMENTS AT T=0 ARE',I7)
90      FORMAT (1HO,'TOTAL REQUIREMENTS FOR SPECIALTIES 10 - 15 BEFORE ANY
       1REDISTRBUTION')
95       FORMAT(1HO,'NUMBER OF RECORDS BYPASSED - ',I5)
97       FORMAT(1HO,'**FATAL ERROR = UNABLE TO READ RECORD WITH INTEGER',
       *'OR A/N FORMAT - PROGRAM TERMINATED')
        NI=0
        IREC=0
C
C       ZERO OUT TOTAL REQUIREMENTS MATRIX
C
C       READ IN BEGINNING YEAR,MONTH,DAY FOR SACS REQUIREMENT DATA
        READ (5,70) ITIME(1)
C       COMPUTE TEN 1-YEAR TIME INTERVALS
        DO 115 K=2,10
        J=K-1
        ITIME(K)=ITIME(J)+10000
```

```
  115 CONTINUE
C
C     READ IN SACS REQUIREMENTS
C
  100 READ (10,10,END=9000,ERR=8999) IGR,ISPEC,IEDATE,ITDATE,IAUTH
      IF (ISPEC .LT. 1) GO TO 100
      DO 120 J=1,10
      IF (ITIME(J) .LT. IEDATE .OR. ITIME(J) .GT. ITDATE) GO TO 120
C     ADD IN AUTHORIZED STRENGTHS
      ITREQ(J,IGR,ISPEC)=ITREQ(J,IGR,ISPEC)+IAUTH
  120 CONTINUE
      GO TO 100
C        RE-READ WITH A/N FORMAT
8999     READ(0,15,END=9000,ERR=9999)IGR,ISPEC,IEDATE,ITDATE,IAUTH
         IBYPAS=IBYPAS+1
C        BYPASS NMT 200 RECORDS - THEN TERMINATE RUN
         IF(IBYPAS .GT. 200) GO TO 400
          WRITE(6,12)
         WRITE(6,15) IGR,ISPEC,IEDATE,ITDATE,IAUTH
          GO TO 100
 9000 CONTINUE
C
C     PRINTOUT TOTAL REQUIREMENTS FOR SPECIALTIES 10-15 BEFORE ANY
C     REDISTRIBUTION
C
      PRINT 90
        DO 590 I=10,15
      PRINT 20, I
      PRINT 80, ITIME(1)
      PRINT 30, (NL,NL=2,6)
      DO 595 J=1,10
      N=J-1
      PRINT 40, N,(ITREQ(J,K,I),K=2,6)
  595 CONTINUE
      DO 599 J=2,6
      ITREQ(11,J,I)=0
      DO 599 K=1,10
      ITREQ(11,J,I)=ITREQ(11,J,I)+ITREQ(K,J,I)
  599 CONTINUE
      PRINT 50, (ITREQ(11,J,I),J=2,6)
  590 CONTINUE
C     BREAKOUT SPECIALTY 10
C
      DO 130 J=1,10
      DO 130 K=2,6
      ITREQ(J,K,11)=ITREQ(J,K,11)+(45*ITREQ(J,K,10))/100
      ITREQ(J,K,12)=ITREQ(J,K,12)+(18*ITREQ(J,K,10))/100
      ITREQ(J,K,13)=ITREQ(J,K,13)+(27*ITREQ(J,K,10))/100
      ITREQ(J,K,14)=ITREQ(J,K,14)+(10*ITREQ(J,K,10))/100
  130 CONTINUE
C
C     BREAKOUT SPECIALTY 08
C
      DO 135 J=1,10
      DO 135 K=2,6
      ITREQ(J,K,42)=ITREQ(J,K,42)+ITREQ(J,K,8)
  135 CONTINUE
```

```
C
C     BREAKOUT SPECIALTY 34
C
      DO 140 J=1,10
      DO 140 K=2,6
      ITREQ(J,K,35)=ITREQ(J,K,35)+ITREQ(J,K,34)
  140 CONTINUE
C
C     BREAKOUT SPECIALTY 40
C
      DO 145 J=1,10
      DO 145 K=2,6
      ITREQ(J,K,41)=ITREQ(J,K,41)+(75*ITREQ(J,K,40))/100
      ITREQ(J,K,42)=ITREQ(J,K,42)+(25*ITREQ(J,K,40))/100
  145 CONTINUE
C
C     BREAKOUT SPECIALTY 24
C
      DO 150 J=1,10
      DO 150 K=2,6
      ITREQ(J,K,25)=ITREQ(J,K,25)+ITREQ(J,K,24)
  150 CONTINUE
C
C     BREAKOUT SPECIALTY 07
C
      DO 155 J=1,10
      DO 155 K=2,6
      ITREQ(J,K,15)=ITREQ(J,K,15)+ITREQ(J,K,7)
  155 CONTINUE
C
C     BREAKOUT SPECIALTY 20
C
      DO 160 J=1,10
      DO 160 K=2,6
      ITREQ(J,K,21)=ITREQ(J,K,21)+ITREQ(J,K,20)
  160 CONTINUE
C
C     BREAKOUT SPECIALTY 22
C
      DO 165 J=1,10
      DO 165 K=2,6
      ITREQ(J,K,21)=ITREQ(J,K,21)+ITREQ(J,K,22)
  165 CONTINUE
C
C     BREAKOUT SPECIALTY 23
C
      DO 170 J=1,10
      DO 170 K=2,6
      ITREQ(J,K,21)=ITREQ(J,K,21)+ITREQ(J,K,23)
  170 CONTINUE
C
C     BREAKOUT SPECIALTY 98 FOR GRADES O-5 THRU O-6
C
      DO 175 J=1,10
      DO 175 K=5,6
      ITREQ(J,K,70)=ITREQ(J,K,70)+ITREQ(J,K,98)
       ITREQ(J,K,98)=0
```

```
  175 CONTINUE
C
C       BREAKOUT SPECIALTY 98 FOR GRADES O-1 THRU O-4
C
        DO 180 J=1,10
        DO 180 K=2,4
        ITREQ(J,K,92)=ITREQ(J,K,92)+ITREQ(J,K,98)
          ITREQ(J,K,98)=0
  180 CONTINUE
C
C       BREAKOUT SPECIALTY 99
C
        DO 185 J=1,10
        DO 185 K=2,6
        ITREQ(J,K,92)=ITREQ(J,K,92)+ITREQ(J,K,99)
          ITREQ(J,K,99)=0
  185 CONTINUE
C
C       PRINTOUT TOTAL REQUIREMENTS BY SPECIALTIES
C
        DO 190 I=1,50
        JN=NSPEC(I)
        IF (JN .EQ. 00) GO TO 190
        PRINT 20, JN
        PRINT 80, ITIME(1)
        PRINT 30, (NL,NL=2,6)
        DO 195 J=1,10
        N=J-1
        PRINT 40, N,(ITREQ(J,K,JN),K=2,6)
  195 CONTINUE
        DO 199 J=2,6
        ITREQ(11,J,JN)=0
        DO 199 K=1,10
        ITREQ(11,J,JN)=ITREQ(11,J,JN)+ITREQ(K,J,JN)
  199 CONTINUE
        PRINT 50, (ITREQ(11,J,JN),J=2,6)
  190 CONTINUE
C
C           REDISTRIBUTION OF SPECIALTY 70 TO LOG SPECIALTIES
         DO 250 J=5,6
         DO 240 N=1,10
         ISUM=0
C         SUM THE LOG SPECIALTY REQUIREMENTS BY GRADE AND YEAR
         DO 230 K=1,50
         ISPEC=NSPEC(K)
         IF(ISPEC.LT.71)GO TO 230
C         LOCATED A LOG SPECIALTY
         IF((ITREQ(N,J,ISPEC).EQ.0).AND.(J.EQ.6))
       * ISUM=ISUM+ITREQ(N,5,ISPEC)/3 + .5
         ISUM=ISUM+ITREQ(N,J,ISPEC)
230      CONTINUE
            SUM=ISUM
          PRORATE SPECIALTY 70  REQUIREMENTS ACCORDING TO RELATIVE STRENGTH
C        DO 235 K=1,50
         IF(NSPEC(K).LT.71) GO TO 235
         ISPEC=NSPEC(K)
         IF (ITREQ(N,J,ISPEC).EQ.0).AND.(J.EQ.6))ITREQ(N,J,ISPEC)=
```

```
     * (((TREQ(N,5,ISPEC)/3)/SUM) * ITREQ(N,J,70)
      IF((ITREQ(N,J,ISPEC).EQ.0).AND.(J.EQ.6))GO TO 235
      FRACT= (ITREQ(N,J,ISPEC)/SUM) * ITREQ(N,J,70)
      ITREQ(N,J,ISPEC)=ITREQ(N,J,ISPEC+FRAC) + 0.5
235   CONTINUE
      ITREQ(N,J,70)=0
240   CONTINUE
250    CONTINUE
C
C     PRINTOUT TOTAL REQUIREMENTS FOR SPECIALTIES 71 - 97
C
      DO 290 I=1,50
      JN=NSPEC(I)
      IF (JN .LT. 71) GO TO 290
      PRINT 20, JN
       WRITE(6,75)
      PRINT 80, ITIME(1)
      PRINT 30, (NL,NL=2,6)
      DO 295 J=1,10
      N=J-1
      PRINT 40, N,(ITREQ(J,K,JN),K=2,6)
  295 CONTINUE
      DO 299 J=2,6
      ITREQ(11,J,JN)=0
      DO 299 K=1,10
      ITREQ(11,J,JN)=ITREQ(11,J,JN)+ITREQ(K,J,JN)
  299 CONTINUE
      PRINT 50, (ITREQ(11,J,JN),J=2,6)
  290 CONTINUE
C
C
C     PRINTOUT TOTAL REQUIREMENTS BY GRADE
C
      DO 300 IGRADE=6,2,-1
      PRINT 21, IGRADE
      PRINT 80, ITIME(1)
      PRINT 31, (NL,NL=1,9)
      DO 310 J=1,50
      JN=NSPEC(J)
      IF (JN .EQ. 00) GO TO 310
      PRINT 41, JN,(ITREQ(IYR,IGRADE,JN),IYR=1,10)
  310 CONTINUE
C        COMPUTE TOTAL REQ. FOR EACH GRADE AND YEAR,ALL SPECIALTIES
      DO 315 IYR=1,10
      ITREQ(IYR,1,100)=0
      DO 315 J=1,50
      JN=NSPEC(J)
      IF (JN .EQ. 00) GO TO 315
      ITREQ(IYR,1,100)=ITREQ(IYR,1,100)+ITREQ(IYR,IGRADE,JN)
  315 CONTINUE
C        COMPUTE TOTAL T=0 REQUIREMENTS
        TOTREQ=TOTREQ+ITREQ(1,1,100)
      PRINT 51, (ITREQ(IYR,1,100),IYR=1,10)
  300 CONTINUE
       WRITE(8,85) TOTREQ
400    WRITE (11) ITREQ
      PRINT 60, IREC,N1
```

```
      WRITE(6,95)IBYPAS
      ENDFILE 11
      STOP FINISH
C       UNABLE TO READ RECORD WITH I OR A FORMAT
9999  WRITE(6,97)
      GO TO 400
      END
```

BEST AVAILABLE COPY

```
C       ****************************************************************
C       *                                                              *
C       *       PROGRAM    : SACSEXTRACT                               *
C       *       CALLED BY: NONE                                        *
C       *       CALLING ARGUMENTS: NONE                                *
C       *       CALLED ROUTINES: ERTRAN                                *
C       *       INPUT FILES:                                           *
C       *                                                              *
C       *                                                              *
C       *       OUTPUT FILES:                                          *
C       *                                                              *
C       *       DATE: 15 APRIL 75                                      *
C       *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C       *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C       *                8120 WOODMONT AVENUE                          *
C       *                BETHESDA,MARYLAND 20014                       *
C       *                                                              *
C       ****************************************************************
        IMPLICIT INTEGER (A-Z)
        DIMENSION IN(1203),OUT(13),IASG(7),NTAPEX(6)
         DATA IASG/42H@ASG,T     10,16D////EBCDIC//6,        .  /
        COMMON /OPTION/IWRITE
C
C        @ASG,THIE  TAPEIN.,8C,SXXXR
C        @ASG,T       TAPEOUT.,8C9,SAVEW
C        @USE 10,TAPEIN.
C        @USE 11,TAPEOUT.
C
        READ(5,930) NTAPE
        WRITE(6,932)NTAPE
        READ(5,931)(NTAPEX(I),I = 1,NTAPE)
        WRITE(6,933)(NTAPEX(I),I= 1,NTAPE)
        GO TO 2
C
1        CALL NTRAN(10,2,1203,IN,L,22)
         IF(L.EQ.-2) GO TO 2
        IZ = 0
        ICTBLK= ICTBLK + 95
        DO 100 IJ = 1,95,3
        IZ = IZ + 1
        IJ1= (IZ-1)*38  + 1
         DECODE(76,911,IN(IJ1))OUT
        CALL FIX(OUT)
        IJ2=IJ1+12
         DECODE(80,912,IN( IJ2))X,OUT
        CALL FIX(OUT)
        IF((IJ+2).GT.95) GO TO 1
        IJ3=IJ1+25
         DECODE(76,913,IN( IJ3))X,OUT
        CALL FIX(OUT)
  911  FORMAT(12A6,A4)
```

```
  912 FORMAT(A4,12A6,A4)
  913 FORMAT (A2,12A6,A4)
 930    FORMAT()
 931    FORMAT(6A6)
 932    FORMAT(' NUMBER OF TAPES BEING READ EQUALS ',I4)
 933    FORMAT(' TAPE NUMBERS ARE ',6(A6,',',2X))
 100    CONTINUE
         GO TO 1
 2       CONTINUE
        ITAPE = ITAPE + 1
        IF (ITAPE.GT.NTAPE)GO TO 30
        IF(ITAPE.GT.1)  CALL ERTRAN(6,'@FREE 10. . ')
        IASG(6) = NTAPEX(ITAPE)
        CALL ERTRAN(6,IASG)
        GO TO 1
 30      ENDFILE 11
         ENDFILE 11
        WRITE(6,940)ICTBLK,IWRITE
 940    FORMAT(' NUMBER OF RECORDS PROCESSED EQUALS ',I8,/
       * ' NUMBER OF RECORDS EXTRACTED EQUALS ',I8)
         STOP
        SUBROUTINE FIX(IOUT)
        COMMON /OPTION/IWRITE
        DIMENSION IOUT(13)
        FLD(0,6,ICOMP)=FLD(30,6,IOUT(12))
        IF(ICOMP.NE.'1@@@@@')RETURN
 C
        FLD(0,6,IOFF)=FLD(6,6,IOUT(4))
        IF(IOFF.NE.'0@@@@@')RETURN      @ TEST FOR OFFICER RECORD
 C
        IGRADE = FLD(12,6,IOUT(4))        @ EXTRACT GRADE
        IF(IGRADE.GT.54)RETURN            @ CONSIDER COL-LT ONLY
 C
        IEDATE = FLD(0,36,IOUT(2))        @ START DATE
        ITDATE = FLD(0,36,IOUT(3))        @ TERMINAL DATE
 C
        IPRMY = FLD(18,12,IOUT(4))        @ EXTRACT PRIMARY SPECIALTY
        IKEY = FLD(30,6,IOUT(4))        @ EXTRACT PRIMARY SPECIALTY KEY
        IF(IKEY.GT.47) IPRMY = FLD(24,12,IOUT(7))
        IAUTH = FLD(0,24,IOUT(10))        @ AUTHORIZED QUANTITY
        FLD(6,6,IAUTH) = FLD(30,6,IOUT(9))       @ AUTHORIZED QTY
        IWRITE = IWRITE + 1
        WRITE(11,902)IGRADE,IPRMY,IEDATE,ITDATE,IAUTH
 902    FORMAT(1X,R1,R2,15X,A6,A6,9X,R5)
 901    FORMAT(1X,12A6,A4)
        RETURN
        END
```

```
C      *****************************************************************
C      *                                                               *
C      *       PROGRAM: SACSPREPRO                                     *
C      *       INPUT FILES: TEMPDISC - COPY OF SACS EXTRACT TAPE (10)  *
C      *       OUTPUT FILES: ODOUTUDOI. (TEMP) - 11 -                  *
C      *       PURPOSE: THIS PROGRAM CORRECTS MISTAKES IN SPECIALTY    *
C      *       NUMBERS ON THE ORIGINAL SACS DATA AND CREATES A USEABLE *
C      *       FILE TO BE USED BY THE PFODSAP .SACSCREATE PROGRAM.     *
C      *       DATE: 15 APRIL 76                                       *
C      *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C      *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C      *                8120 WOODMONT AVENUE                           *
C      *                BETHESDA,MARYLAND 20014                        *
C      *                                                               *
C      *****************************************************************
       DIMENSION IOUT(8)
       COMMON DBLZRO,ONEBLK,FOURBK,FIVEA,SIXBLK,SIXA,THRS,THRAB,
      1THRCD,THREF,THRGH,THRLP,THRUV,THRQ,THRW
       INTEGER DBLZRO,ONEBLK,FOURBK,FIVEA,SIXBLK,SIXA,THRS,THRAB,
      1THRCD,THREF,THRGH,THRLP,THRUV,THRQ,THRW
    3  FORMAT(8A6)
 42      FORMAT(1H0,'DUBLZERO = ',I8,' ONE-BLANK = ',I8,
      &' FOUR-BLANK = ',I8/1H0,' FIVE-A = ',I8,' SIX-BLANK = ',I8,
      &' SIX-A = ',I8/1H0,' THREE-S = ',I8,' THREE-AB = ',I8,
      &' THREE-CD = ',I8/1H0,' THREE-EF = ',I8,' THREE-GH = ',I8,
      &'THREE-LP = ',I8/1H0,' THREE-UV = ',I8,' THREE-Q = ',I8,
      &' THREE-W = ',I8)
 45      FORMAT(1H0,'ISUM = ',I10)
 2000 FORMAT ('1  END PROGRAM: RECORDS = 'I6777)
       ITOTAL=0
 1     READ(10,3,END=9999)IOUT
       ITOTAL= ITOTAL+1
       CALL PREPRO(IOUT)
       GO TO 1
 9999 END FILE 11
 9000 PRINT 2000, ITOTAL
         WRITE (6,42)DBLZRO,ONEBLK,FOURBK,FIVEA,SIXBLK,SIXA,
      *THRS,THRAB,THRCD,THREF,THRGH,THRLP,THRUV,THRQ,THRW
        ISUM=DBLZRO+ONEBLK+FOURBK+FIVEA+SIXBLK+SIXA+THRS+THRAB
      *       +THRCD+THREF+THRGH+THRLP+THRUV+THRQ+THRW
         WRITE(6,45)ISUM
       REWIND 11
       STOP FINISH
C
C      *** INTERNAL SUBROUTINE FOR READING AND RE-FORMATTING
C      *** SACS EXTRACT TAPE ***
C
       SUBROUTINE PREPRO(IOUT)
       DIMENSION IHAY(10),IOUT(8)
       COMMON DBLZRO,ONEBLK,FOURBK,FIVEA,SIXBLK,SIXA,THRS,THRAB,
      1THRCD,THREF,THRGH,THRLP,THRUV,THRQ,THRW
       INTEGER DBLZRO,ONEBLK,FOURBK,FIVEA,SIXBLK,SIXA,THRS,THRAB,
      1THRCD,THREF,THRGH,THRLP,THRUV,THRQ,THRW
   10 FORMAT (A2,A1,R1,6A6,A5,3X)
   30 FORMAT (A2,A1,R1,6A6,A5)
```

```
      DECODE(48,10,IOUT) (IRAY(J),J=1,10)
      IF ((IRAY(2).EQ.'0'.AND.IRAY(3).EQ.'@@@@@0') GO TO 777
      IF ((IRAY(3) .GE. 48 .AND. IRAY(3) .LE. 57) GO TO 200
C
C     MISTAKE FOUND IN RECORD
C     IRAY(2) CONTAINS FIRST DIGIT OF SPECIALTY NUMBER
C     IRAY(3) CONTAINS SECOND DIGIT OF SPECIALTY NUMBER
      IF ((IRAY(2) .EQ. '1') GO TO 1
      IF ((IRAY(2) .EQ. '2') GO TO 1
      IF ((IRAY(2) .EQ. '3') GO TO 3
      IF ((IRAY(2) .EQ. '4') GO TO 4
      IF ((IRAY(2) .EQ. '5') GO TO 5
      IF ((IRAY(2) .EQ. '6') GO TO 6
C
C     SET UNKNOWN SPECIALTIES TO DOUBLE ZERO
      IRAY(2)='0'
      IRAY(3)=IR0
777    DBLZRO=DBLZRO+1
        RETURN
C
C     CONVERT TO SPECIALTY 15
    1 IRAY(2)='1'
      IRAY(3)=IR5
      ONEBLK=ONEBLK+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 48
    4 IRAY(2)='4'
      IRAY(3)=IR8
        FOURBK=FOURBK+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 31
    5 IF ((IRAY(3) .EQ. IRA) IRAY(2)='1'
      IF ((IRAY(2) .NE.'1') IRAY(2)='3'
      IRAY(3)=IR1
        FIVEA=FIVEA+1
      GO TO 200
    6 IF ((IRAY(3) .EQ. IRA) GO TO 7
C
C     CONVERT TO SPECIALTY 25
      IRAY(2)='2'
      IRAY(3)=IR5
        SIXBLK=SIXBLK+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 36
    7 IRAY(2)='3'
      IRAY(3)=IR6
       SIXA=SIXA+1
      GO TO 200
C
C     CHECK SECOND DIGIT OF SPECIALTY 3 -
    3 IF ((IRAY(3) .EQ. IRA .OR. IRAY(3) .EQ. IRB) GO TO 13
      IF ((IRAY(3) .EQ. IRC .OR. IRAY(3) .EQ. IRD) GO TO 14
      IF ((IRAY(3) .EQ. IRE .OR. IRAY(3) .EQ. IRF) GO TO 15
      IF ((IRAY(3) .EQ. IRG .OR. IRAY(3) .EQ. IRH) GO TO 16
```

```
      IF (IRAY(3) .EQ. IRL .OR. IRAY(3) .EQ. IRM) GO TO 17
      IF (IRAY(3) .EQ. IRN .OR. IRAY(3) .EQ. IRP) GO TO 17
      IF (IRAY(3) .EQ. IRU .OR. IRAY(3) .EQ. IRV) GO TO 18
      IF (IRAY(3) .EQ. IRQ) GO TO 19
      IF (IRAY(3) .EQ. IRW) GO TO 21
      IF (IRAY(3) .EQ. IRS) IRAY(3)=IR7
      IRAY(2)='9'
         THRS=THRS+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 53
   13 IRAY(2)='5'
      IRAY(3)=IR3
        THRAB=THRAB+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 52
   14 IRAY(2)='5'
      IRAY(3)=IR2
         THRCD=THRCD+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 45
   15 IRAY(2)='4'
      IRAY(3)=IR5
         THREF=THREF+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 46
   16 IRAY(2)='4'
      IRAY(3)=IR6
        THRGH=THRGH+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 99
   17 IRAY(2)='9'
      IRAY(3)=IR9
        THRLP=THRLP+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 51
   18 IRAY(2)='5'
      IRAY(3)=IR1
       THRUV=THRUV+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 49
   19 IRAY(2)='4'
      IRAY(3)=IR9
        THRQ=THRQ+1
      GO TO 200
C
C     CONVERT TO SPECIALTY 37
   21 IRAY(2)='3'
      IRAY(3)=IR7
       THRW=THRW+1
C     WRITE OUT CORRECTED SPECIALTY
  200 WRITE (11,30) (IRAY(J),J=1,10)
      RETURN
      END
```

```
C
C     ..........................................................................
C     ..........................................................................
C     .                                                                        .
C     .        PROGRAM: .TOURATIOS                                             .
C     .        INPUT FILES: ODTURUD01.                                         .
C     .                     ODRATUD01.(ONLY IN UPDATE MODE)                    .
C     .        OUTPUT FILES: ODRATUD01.                                        .
C     .        PURPOSE: THIS PROGRAM CREATES AND UPDATES UTILIZATION           .
C     .        RATIOS AND TOUR LENGTHS BY SPECIALTY PAIRINGS FOR GRADES        .
C     .        O-6 THRU O-4, AND COMPUTES NUMBER OF PREFERENCES(NPREF)         .
C     .        DATE: 15 APRIL 76                                               .
C     .        AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN          .
C     .        COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                     .
C     .                 8120 WOODMONT AVENUE                                   .
C     .                 BETHESDA,MARYLAND 20014                                .
C     .                                                                        .
C     ..........................................................................
      COMMON ISPEC(50)
      DIMENSION UTLR8(12),IUTLR8(11),IUTIL(6),UTIL(6),NPROB(10)
      INTEGER TOUR(6),URATIO(50,50,6)
      DATA UTLR8/0.,.18,.22,.28,.38,.46,.54,.66,.70,.78,.82,1.0/
      DATA IUTLR8/15,14,13,12,23,11,32,21,31,41,51/
      DATA ISPEC/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
     145,46,47,48,49,51,52,53,54,71,72,73,74,75,76,77,81,82,83,86,87,88,
     291,92,93,95,97,00,00,00,00,00/
      DEFINE FILE 10(6,2500,U,IREC)
    1 FORMAT (2I2,3(F5.3,I2))
    2 FORMAT (A6)
    4 FORMAT(1H1,'NO. OF CARDS READ =',I5/1X,'NO. OF PREFERENCES =',I5)
    5 FORMAT (1H1,'INCORRECT MODE NAME ENTERED')
    6 FORMAT ( )
    7 FORMAT (1X,'CARD NO.',I4,' HAS A PROHIBITED  ALTERNATE SPECIALTY'/
     A1X,'ALL RATIOS AND TOUR LENGTHS WERE IGNORED')
    8     FORMAT(1H0,'FOLLOWING INPUT RECORD ATTEMPTS TO REDEFINE A ',
     1'TOUR LENGTH FOR PRIMARY SPECIALTY ',I3,'.'/'0 PREVIOUSLY ',
     2'DEFINED TOUR LENGTH IS ',I3,'.',' THE TOUR LENGTH WAS NOT ',
     3'UPDATED.'//1H0,2I2,3(F5.3,I2),' CARD NBR ',I7)
    9     FORMAT(1H0,'NO TOUR LENGTH INPUT FOR SPECIALTY ',I3/1H0,
     A'DEFAULT VALUE OF ..3.. WAS INSERTED FOR ALL GRADES')
   20     FORMAT(1H0,'CARD NUMBER ',I4,' IS A DUPLICATE'/
     *1H0,'CARD IMAGE WAS ',2I2,3(F5.3,I2)/
     *1H0,'IT WAS REJECTED')
C     READ IN WHETHER TO UPDATE OR TO CREATE INITIAL TOURATIO FILE
C     AND READ IN PROHIBITED ALTERNATE SPECIALTIES
      READ (5,2) MODE
      READ (5,6) NBRPRO,(NPROB(J),J=1,NBRPRO)
      IF (MODE.EQ.'CREATE'.OR.MODE.EQ.'UPDATE')GO TO 70
      GO TO 8000
   70 NCARD=0
      NPREF=0
      IF (MODE .EQ. 'UPDATE') GO TO 101
C
C     INITIALIZE UTILIZATION RATIOS
      DO 91 J=6,4,-1
```

```
      IREC=7-J
      DO 76 M=1,50
      DO 75 N=1,50
      URATIO(M,N,IREC)=99
75    CONTINUE
76    CONTINUE
C     SET CODE FOR PROHIBITED ALTERNATE SPECIALTY
      DO 77 M=1,50
      DO 74 N=1,50
      DO 73 K=1,NBRPRO
      IF (NPROB(K) .NE. ISPEC(N)) GO TO 73
      URATIO(M,N,IREC)=88
      GO TO 74
73    CONTINUE
74    CONTINUE
77    CONTINUE
   91 CONTINUE
      GO TO 100
C     READ IN UTILIZATION RATIOS AND TOUR LENGTHS
  101 DO 198 IREC=1,6
 198   READ (10'IREC) ((URATIO(J,K,IREC),J=1,50),K=1,50)
C
C     ****** CREATE TOURATIO FILE ******
C
C     READ IN PRIMARY,ALTERNATE,UTILRATIO,AND TOUR LENGTH
  100 READ (5,1,END=9000,ERR=100) IPR,IALT,(UTIL(K),TOUR(K),K=6,4)
      NCARD=NCARD+1
      IVAL=0
C     CHECK TO SEE IF ALT. SPECIALTY IS A COMBAT ARMS
      DO 105 J=1,NBRPRO
      IF(IALT.EQ.NPROB(J))WRITE(6,7)NCARD
      IF (IALT .EQ. NPROB(J)) GO TO 100
105   CONTINUE
C     CHECK FOR VALID SPECIALTY(PRIMARY)
      CALL VALID(IPR,NCARD,IVAL,IPNTR)
C     SAVE POINTER FOR PRIMARY SPECIALTY
      IPRI=IPNTR
C     CHECK FOR VALID SPECIALTY(ALTERNATE)
      CALL VALID(IALT,NCARD,IVAL,IPNTR)
C     SAVE POINTER FOR ALTERNATE SPECIALTY
      IALTI=IPNTR
      IF (IVAL .EQ. 99) GO TO 100
C         DUPLICATE  CARD CHECK
        IF(MODE .EQ. 'UPDATE') GO TO    110
         IF(URATIO(IPRI,IALTI,1).GE.88.)
     * GO TO 110
C          CHECK IF A TOUR LENGTH ENTRY
       IF (URATIO(IPRI,IPRI ,6).EQ.0.) GO TO 110
      WRITE(6,20)NCARD,IPR ,IALT ,(UTIL(K),TOUR(K),K=6,4)
       GO TO 100
C     DETERMINE INTEGER VALUE FOR UTILRATIO
110   DO 90 K=4,6
      U=UTIL(K)
      IF (U .GT. 1.0) U=.9
      DO 80 J=1,12
      IF (ABS(U) .GT. UTLRB(J) .AND. ABS(U) .LE. UTLRB(J+1)) GO TO 79
   80 CONTINUE
```

```
   79 IUTIL(K)=IUTLR8(J)
   90 CONTINUE
C       CREATE UTILIZATION RATIOS
        DO 95 IGRADE=4,6
        IREC=7-IGRADE
C       CHECK FOR PROHIBITED ALTERNATE SPECIALTY
        DO 71 J=1,NBRPRO
C       IF PRIMARY SPECIALTY IS A PROHIBITED ALT-U.R. IS AS GIVEN
        IF (IPR .EQ. NPROB(J)) GO TO 94
71      CONTINUE
C       TEST TO SEE IF ALT/PRIMARY PREFERENCE SHOULD BE CREATED
        MRATIO=IUTIL(IGRADE)/10
        NRATIO=IUTIL(IGRADE)-(10*MRATIO)
        IF (IALT .LE. IPR) GO TO 94
        IF (URATIO(IALT,IPR,IREC) .NE. 99) GO TO 94
        URATIO(IALT,IPR,IREC)=(NRATIO*10)+MRATIO
   94 URATIO(IPR,IALT,IREC)=IUTIL(IGRADE)
        IF (URATIO(IALT,IPR,IREC) .NE. 99) GO TO 95
        URATIO(IALT,IPR,IREC)=(NRATIO*10)+MRATIO
   95 CONTINUE
C
C       CREATE TOUR LENGTHS
C
        DO 97 IGRADE=4,6
        IREC=10-IGRADE
C         IS THIS THE FIRST TIME THAT THE TOUR LENGTH FOR THE
C          SPECIALTY WAS INPUT -- IF YES-ACCEPT
          INDEX=IPR
          IF(URATIO(IPR,INDEX,IREC).LT.1)URATIO(IPR,INDEX,IREC)=
      ATOUR(IGRADE)/10
C         DOES SUBSEQUENT INPUT AGREE WITH EARLIER INPUT--IF YES-
C          CONTINUE.  IF NOT EQUAL--ERROR-INPUT NOT ACCEPTED
         IF((TOUR(IGRADE)/10).EQ.URATIO(IPR,INDEX,IREC))GO TO 97
          WRITE(6,8)IPR,URATIO(IPR,IALT,IREC),IPR,IALT,
      A(UTIL(K),TOUR(K):K=6,4),NCARD
97        CONTINUE
        GO TO 100
C       WRITE OUT ERROR MESSAGE
 8000 WRITE (6,5)
        STOP ERROR
9000    CONTINUE
C         CHECK TO SEE IF ALL SPECIALTIES HAVE A VALID TOUR LENGTH
C         AND MOVE TO COLUMN  ONE
          DO 99 J=6,4,-1
          IREC=10-J
          DO 98 M=1,50
          IF(URATIO(M,M,IREC).GT.0)GO TO 985
C         IF LAST SPECIALTY, THEN STOP THE LOOP
        IF(ISPEC(M).EQ.0) GO TO 99
          WRITE(6,9) ISPEC(M)
          URATIO(M,1,IREC)=3
        NEXT1=IREC+1
        NEXT2=IREC+2
        URATIO(M,M,NEXT1)=3
        URATIO(M,M,NEXT2)=3
          GO TO 98
985       URATIO(M,1,IREC)=URATIO(M,M,IREC)
```

```
98      CONTINUE
99      CONTINUE
C     WRITE OUT UTTILIZATION RATIOS AND TOUR LENGTHS
      DO 199 IREC=1,6
      WRITE(6,901)IREC
901   FORMAT('1 TOUR RATIO FILE RECORD NUMBER ',I4)
      WRITE (6,902) ((URATIO(J,K,IREC),K=1,50),J=1,50)
902   FORMAT(5X,50I2)
199     WRITE (10'IREC) ((URATIO(J,K,IREC),J=1,50),K=1,50)
C         COUNT THE NUMBER OF PREFERENCES
          DO  9600  J =1,50
          DO 9500   K =1,50
        IF(URATIO(J,K,1).GT.88.) GO TO 9500
C         IF BOTH COMBAT ARMS BYPASS
          IF((URATIO(J,K,1).EQ.88.).AND.(URATIO(K,J,1).GE.88.))
     * GO TO 9500
C         IF A PREFERENCE OF A COMBAT ARM COUNT TWICE
          IF((URATIO(J,K,1).EQ.88.).AND.(URATIO(K,J,1).LT.88.))
     * NPREF=NPREF+2
C         FOR PREFERENCES OF TWO NON-COMBAT ARMS
          IF((URATIO(J,K,1).LT.88.).AND.(URATIO(K,J,1).EQ.88.))
     * GO TO 9500
          IF((URATIO(J,K,1).LT.88.).AND.(URATIO(K,J,1).NE.88.))
     * NPREF=NPREF+1
9500      CONTINUE
9600      CONTINUE
      PRINT 4, NCARD,NPREF
      STOP FINISH
C
C     INTERNAL SUBROUTINE VALID
C
      SUBROUTINE VALID(M,ICARD,NVAL,IPTR)
      COMMON ISPEC(50)
    3 FORMAT (1X,'CARD NO.',I4,' HAS INVALID SPECIALTY NO.',I3)
      DO 1000 J=1,50
      IF (M .NE. ISPEC(J)) GO TO 1000
      IPTR=J
      GO TO 2000
 1000 CONTINUE
      WRITE (6,3) ICARD,M
      NVAL=99
 2000 RETURN
      END
```

```
      BLOCK DATA
C     ***************************************************************
C     *                                                             *
C     *       SUBROUTINE: BLOCK DATA                                *
C     *       CALLED BY: NONE                                       *
C     *       CALLING ARGUMENTS: NONE                               *
C     *       CALLED ROUTINES: NONE                                 *
C     *       PURPOSE: THIS BLOCKDATA ROUTINE ESTABLISHES THE       *
C     *                VALUES OF THE BASIC DATA AT THE START OF     *
C     *                THE PROGRAM.                                 *
C     *       DATE: 15 APRIL 76                                     *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                8120 WOODMONT AVENUE                         *
C     *                BETHESDA,MARYLAND 20014                      *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1.LIST
C
      DATA MINUS1,PLUS1,ZERO/ -1.,+1.,0.0/
C
      DATA MAXSPC,MAXYRS/50,9/
C
      DATA NAME/'CREQ','TREQ','GOZO','LINC','CINC','UBSG'/
C
      DATA NBRSPC/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
     145,46,47,48,49,51,52,53,54,71,72,73,74,75,76,77,81,82,83,86,87,88,
     291,92,93,95,97,0,0,0,0,0/
C
      DATA NN,RES,RR/'N','RES','R'/
C
      END
```

```
      SUBROUTINE BOUNDS
C     *************************************************************
C     *************************************************************
C     *                                                           *
C     *       SUBROUTINE: BOUNDS                                  *
C     *       CALLED BY: MAIN                                     *
C     *       CALLING ARGUMENTS: NONE                             *
C     *       CALLED ROUTINES: IBITS,IPHASE,VALID                 *
C     *       OUTPUT FILES: 9 = ODEQAUD01                         *
C     *       PURPOSE: THIS SUBROUTINE WRITES THE BOUNDS CHAPTER  *
C     *                FOR THE FMPS MODEL.                        *
C     *       DATE: 15 APRIL 76                                   *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY         *
C     *                8120 WOODMONT AVENUE                       *
C     *                BETHESDA,MARYLAND 20014                    *
C     *                                                           *
C     *************************************************************
      INCLUDE PROC1
      INTEGER ARCID
      REAL LOBND
          PRODRB=1.0
C
C                        *****FORMAT STATEMENTS*****
C
10    FORMAT ('BOUNDS')
20    FORMAT (1X,'LO',1X,'BOUNDS',4X,'X000',J2,4X,F12.3)
30    FORMAT (1X,'UP',1X,'BOUNDS',4X,'XN',J2,6X,F12.3)
40    FORMAT (1X,'LO',1X,'BOUNDS',4X,'XN',J2,6X,F12.3)
50    FORMAT (1X,'FX',1X,'BOUNDS',4X,'X',J5,4X,F12.3)
60    FORMAT (1X,'UP',1X,'BOUNDS',4X,'X000',J2,4X,F12.3)
70    FORMAT (1X,'UP',1X,'BOUNDS',4X,'Y',J1,J2,J2,4X,F12.3)
C
C          ****************************************
      WRITE (9,10)
C
         IF(JGRADE .EQ. 3) GO TO 150
      IF (JGRADE.EQ.2) GO TO 90
      IF (NSPEC.LT.30.OR.ISEG.GT.1) GO TO 130          B SEGMENT
C        ***UPPERBOUND ON X000N - SELECTED SPEC CONTROL OF INPUT
         DO 80 NX = 1,NBRPRO
         DO 81 M  = 1,NSPEC
         IF(NPROB(NX) .EQ. NBRSPC(M)) GO TO 82
81       CONTINUE
      GO TO 80
82       CONTINUE
         INDEX=(M-1)*(NYRS+1)+1
         UPPER= MAX(UPBND(M)- FLOAT(REQ2(INDEX)),0.)
          WRITE (9,60) NBRSPC(M),UPPER
80       CONTINUE
      GO TO 130
C
90       DO 100 M=1,NSPEC
C        ***UPPERBOUND ON ALL SPECIALITIES AT T0,X000-- (LT ONLY)
         INDEX=(M-1)*(NYRS+1)+1
```

```
          UPPER=FLOAT(REQ2(INDEX))
          WRITE (9,60) NBRSPC(M),UPPER
100       CONTINUE
C
          ILAST = MIN(NYRS,ICHG)
          DO 120 J=1,ILAST
          K=J-1
             DO 110 M=1,NSPEC
C         ***UPPERBOUND ON ALL INCUMBENTS FOR TO = [T-ICHG]   (LT ONLY)
             INDEX=(M-1)*(NYRS+1)+1+J
             UPPER=FLOAT(REQ1(INDEX))
C      SCAN REQMTS FOR BLOCKAGES, RESET REQMTS IF A BLOCKAGE
       IF(FLOAT(REQ1(INDEX+1)).LT.UPPER)REQ1(INDEX+1)=REQ1(INDEX)
             WRITE (9,70) K,NBRSPC(M),NBRSPC(M),UPPER
110          CONTINUE
120       CONTINUE
130    IF (JGRADE.LE.3) GO TO 150
C        ****SETTING XO MM=0 IF NO FLOW INTO M****
          DO 140 M=1,NSPEC
          CALL IPHASE(M,$140)                       @ SEGMENT
          INDEX=(M-1)*NSPEC+M
          IF (IBITS(RTYPE,INDEX).NE.3) GO TO 140
          ARCID=NBRSPC(M)*100+NBRSPC(M)
          WRITE (9,50) ARCID,ZERO
140       CONTINUE
C          ***SETTING UB ON XN-- = TREQ IN LAST YEAR
C
150        DO 155 J=1,NYRS
           PRODR8=PRODR8*(MIN(SURVHI(J),SURVLO(J)))
155        CONTINUE
          DO 170 M=1,NSPEC
          CALL IPHASE(M,$170)                       @ SEGMENT CHECK
          INDEX=(M)*(NYRS+1)
          UPPER=FLOAT(REQ2(INDEX))
          IF (JGRADE.NE.3) GO TO 160
C          ***FOR CPTS ONLY - LO BOUND ON XN-- GT 0 FOR AES ONLY
           LOBND=ZERO
C          **SET NON-ZERO LOWER BOUND FOR AES***
           IF(VALID(M).GT.0)LOBND=FLOAT(REQ2(INDEX)*PRODR8)
          WRITE (9,40) NBRSPC(M),LOBND
160       WRITE (9,30) NBRSPC(M),UPPER
170       CONTINUE
       RETURN
       END
```

```
      SUBROUTINE CPDIAG
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *      SUBROUTINE: CPDIAG                                      *
C     *      CALLED BY: HICOL                                        *
C     *      CALLING ARGUMENTS: NONE                                 *
C     *      CALLED ROUTINES: IBITS,VALID                            *
C     *      OUTPUT FILES: 9 - ODEGAUDOI                             *
C     *      PURPOSE: DEFINES VARIABLES,APPROPIATE COEFFICIENTS      *
C     *               AND CONSTRAINTS FOR CPTS/MAJS WITH TWO         *
C     *               SPECIALTIES IN THE CAPTAIN'S SEGMENT.          *
C     *      DATE: 15 APRIL 78                                       *
C     *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *               8120 WOODMONT AVENUE                           *
C     *               BETHESDA,MARYLAND 20014                        *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
      INTEGER ARCID
C
C                          *****FORMAT STATEMENTS*****
C
10    FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J1,J2,A4,2X,
     1F12.9)
20    FORMAT (4X,'Y',J5,4X,'RES',J5,2X,F12.9,3X,A1,J1,J2,A4,2X,F12.9)
30    FORMAT (4X,'XN',J2,6X,'N',J1,J2,A4,2X,F12.9,3X,'OBJECTIV',2X,
     1F12.9)
121   FORMAT('Y',J5,1X,411,J5,14)
C
C     ****************************************************************
      HOLD1=1
      HOLD2=1
C        ***COMPUTING COEFF FOR FLOWS LEAVING SPEC M**
C          **********          *********          *********
         DO 270 J=1,NYRS
         K=J-1
            DO 260 M=1,NSPEC
               DO 250 N=1,NSPEC
               IF (M.EQ.N) GO TO 250
C
C       IF BES ...DO THE FOLLOWING...
               IF (VALID(M).GT.0) GO TO 40
               IF (ICHG.GT.LASTGO(M,N)) LASTGO(M,N)=ICHG
C
40             INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               NITOUR=0
               ARCID=K*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
C     TO CONSIDER THOSE LEAVING SPEC M THE FIRST TIME
               IF (K.GT.LASTGO(M,N)) GO TO 220
C       *****IF NOT  PREFERRED, THEN ARC NOT CREATED***
               IF (YUTIL(N,M).EQ.99) GO TO 250
C        ***IS M A COMBAT ARM  (N,M)=88
               IF (YUTIL(N,M).NE.88) GO TO 50
```

```
C          ***ARE TWO COMBAT ARMS INVOLVED***
                IF (YUTIL(M,N).GE.88) GO TO 250
                MRATIO=YUTIL(M,N)/10
                NRATIO=YUTIL(M,N)-(MRATIO*10)
                GO TO 60
50              NRATIO=YUTIL(N,M)/10
                MRATIO=YUTIL(N,M)-NRATIO*10
C     ENTER THE GOZINTA AND GOZOUTA VALUES
C                 **GOZO**                **GOZO**
60              WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,
     1          NBRSPC(N),NAME(3),SURVHI(J)
C
                IF (J.EQ.NYRS) GO TO 70
C          ***  RESQ M N  ***                         *** TREQ ***
                WRITE (9,20) ARCID,ARCID,MINUS1,NN,J,NBRSPC(N),NAME(2),
     1          PLUS1
                GO TO 80
C                  *** RESQ M N ***
70              WRITE (9,20) ARCID,ARCID,MINUS1
80              CONTINUE
                LENGTH=YTOUR(N)*NRATIO
                KT=K
                IF ((KT+LENGTH).GE.NYRS) GO TO 190
C        ***THERE WILL BE FLOW OUT OF SPEC N IN KT + LENGTH**
                KPER=KT+LENGTH
C      ***HOW MANY CINC CONSTRAINTS IN TOUR LENGTH
                HOLD1=1.0
90              IEND=LENGTH-1
                IF (IEND.GT.0) GO TO 100
C          **ONE YEAR TOUR***
                NITOUR=NITOUR+1
                IF (NITOUR.GT.1) GO TO 180
                KPER2=K
                GO TO 130
100                DO 110 NI=1,IEND,2
                   KPER1=K+NI
                   KPER2=KPER1+1
                   VALUE1=HOLD1*SURVHI(KPER1)
                   VALUE2=VALUE1*SURVHI(KPER2)
                   IF ((NI+1).GT.IEND) GO TO 120
C                  ****CINC**               **CINC****
                   WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1,NN,
     1             KPER2,NBRSPC(N),NAME(5),VALUE2
110                CONTINUE
                HOLD1=VALUE2
                GO TO 130
C                 **CINC***
120             WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1
                HOLD1=VALUE1
                KPER2=KPER1
C         IS THERE TIME TO ROTATE BACK TO SPEC M
130             IF ((KPER2+1).GE.NYRS) GO TO 240
C     RESTRICTIVE FLOW IN PERIOD KPER, N TO M
                INDEX=(KPER*NSPEC**2)+(N-1)*NSPEC+M
                HOLD1=HOLD1*SURVHI(KPER)
C
                IDOUT=KPER*(10000)+NBRSPC(N)*100+NBRSPC(M)
```

```
C       *****RES-N-M******
              WRITE (9,20) ARCID,IDOUT,HOLD1
      IVALIX = HOLD1*1000
      WRITE(8,121) IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
C     ROTATE TO M AND DETERMINE TIME IN SPEC M
              LENGTH=MRATIO*YTOUR(M)
              IF ((KPER+LENGTH-1).GE.NYRS) LENGTH=NYRS-KPER
C      ***FULL TOUR
C     ****HOW MANY CINC CONSTRAINTS FOR SPEC M
              IEND=LENGTH-1
              IF ((IEND.GT.0) GO TO 140
C         ***ONE YEAR TOUR**
              KPER2=KPER2+1
              GO TO 170
140              DO 190 MI=1,IEND,2
                 KPER1=KPER+MI
                 KPER2=KPER1+1
                 VALUE1=HOLD1*SURVM1(KPER1)
                 VALUE2=VALUE1*SURVM1(KPER2)
                 IF ((MI+1).GT.IEND) GO TO 160
C                ***CINC**               **CINC**
                 WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(5),VALUE1,NN,
     1           KPER2,NBRSPC(M),NAME(5),VALUE2
                 HOLD1=VALUE2
190              CONTINUE
              GO TO 170
C                    ***CINC****
160           WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(5),VALUE1
              HOLD1=VALUE1
              KPER2=KPER1+1
C      ****ARE THERE ANY PERIODS BEYOND KPER2
170           IF ((KPER2+1).GE.NYRS) GO TO 240
C     ******* ROTATE TO SPEC N *******
              K=KPER2+1
C         ***ADD THE LENGTH OF M***
              KPER=KPER+LENGTH
              IF (KPER.GE.NYRS) GO TO 240
              IDOUT=(KPER)*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
              HOLD1=HOLD1*SURVM1(KPER)
              VALUE1=HOLD1
C      ***** WRITE OUT RESTRICTIVE FLOW FOR PERIOD KPER
C                                                    ****RES--M-N******
              INDEX=(KPER*NSPEC**2)+(M-1)*NSPEC+N
              WRITE (9,20) ARCID,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
              LENGTH=YTOUR(N)*NRATIO
C          ***ADD THE LENGT OF N***
              KPER=KPER+LENGTH
C     **IS THERE TIME TO STAY IN N FOR 1 MORE YR**
              K=K+1
              IF ((K).LT.NYRS) GO TO 90
              GO TO 240
C        **SPEC N IS ONE YEAR LONG***
180           IDOUT=(KPER*10000)+(NBRSPC(N)*100)+NBRSPC(M)
              HOLD1=HOLD1*SURVM1(KPER)
              VALUE1=HOLD1
```

```
C                                              ***RES--N-M****
            INDEX=(KPER*NSPEC**2)+(N-1)*NSPEC+M
            WRITE (9,20) ARCID,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,NRATIO,MRATIO,YYOUR(N),YYOUR(M),ARCID,IVALIX
            KPER2=KPER
            GO TO 130
C        **TIME IN SPEC N EXCEEDS HORIZON-WILL NOT LEAVE N***
190         IF (K.EQ.(NYRS-1)) GO TO 250
            RLEFT=SURVHI(J)
C                         **CINC***
            WRITE (9,10) ARCID,J,NBRSPC(N),NAME(5),RLEFT
C         **HOW MANY MORE CINCS***
            IEND=NYRS-(K+2)
            IF (IEND.LE.0) GO TO 250
               DO 210 L=1,IEND,2
               KPER1=K+L+1
               KPER2=KPER1+1
               VALUE1=RLEFT*SURVHI(KPER1)
               VALUE2=VALUE1*SURVHI(KPER2)
               IF ((L+1).GT.IEND) GO TO 200
C                ***CINC**                  ***CINC***
               WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1,NN,
     1         KPER2,NBRSPC(N),NAME(5),VALUE2
               RLEFT=VALUE2
               GO TO 210
C             ***CINC***
200            WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1
210            CONTINUE
            GO TO 250
C        ***K .GT. LASTGO(M)  ***
C      ***IF THERE IS NO RESTRICTIVE FLOW, THEN ARC NOT CREATED***
220         IF (IBITS(RESFLO,INDEX).EQ.0) GO TO 250
C                **GOZO**                        **GOZO**
            WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,
     1      NBRSPC(N),NAME(3),SURVHI(J)
            IDOUT=K*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
            INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
            IF (J.NE.NYRS) GO TO 230
C                                  ******RES--M-N*****
            WRITE (9,20) ARCID,IDOUT,MINUS1
            GO TO 250
230         CONTINUE
C             **RES--M-N**                      ***TREQ***
            WRITE (9,20) ARCID,IDOUT,MINUS1,NN,J,NBRSPC(N),NAME(2),
     1      PLUS1
            GO TO 250
C                                               ***TREQ***
240         K=KT
250         CONTINUE
260      CONTINUE
270   CONTINUE
C     ***********          *******          ********
C     ***********          *******          ********
C         ******WRITE CONSTRAINTS FOR XN******
C
      DO 280 J=1,NSPEC

C      *******N MMGOZO*****          *****OBJECTIV****
      WRITE (9,30) NBRSPC(J),NYRS,NBRSPC(J),NAME(3),MINUS1,PLUS1
280   CONTINUE
C     ********           *******          *********
      RETURN
      END
```

```
      SUBROUTINE HICOL
C     ******************************************************************
C     *                                                                *
C     *       SUBROUTINE: HICOL                                        *
C     *       CALLED BY: MAIN                                          *
C     *       CALLING ARGUMENTS: NONE                                  *
C     *       CALLED ROUTINES: CPDIAG,HICOLS                           *
C     *       PURPOSE: DETERMINES WHICH OF TWO SUBROUTINES TO USE      *
C     *                TO DEFINE VARIABLES FOR THOSE PROMOTED,         *
C     *                DEPENDING UPON GRADE SEGMENT.                   *
C     *       DATE: 15 APRIL 78                                        *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *                8120 WOODMONT AVENUE                            *
C     *                BETHESDA,MARYLAND 20014                         *
C     *                                                                *
C     ******************************************************************
      INCLUDE PROC1
      IF (JGRADE.EQ.3) CALL CPDIAG
      IF (JGRADE.NE.3) CALL HICOLS
      RETURN
      END
```

```
      SUBROUTINE HICOLS
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *       SUBROUTINE: HICOLS                                     *
C     *       CALLED BY: HICOL                                       *
C     *       CALLING ARGUMENTS: NONE                                *
C     *       CALLED ROUTINES: IBITS,IPHASE,JPHASE                   *
C     *       OUTPUT FILES: - 9 - ODEQAUD01.                         *
C     *       PURPOSE: DEFINES VARIABLES,APPROPRIATE COEFFICIENTS    *
C     *                AND CONSTRAINTS FOR PROMOTEES IN LTC/MAJ      *
C     *                SEGMENT. IN LT SEGMENT,DEFINES VARIABLES      *
C     *                REPRESENTING CPTS W/GT SEVEN YEARS OF SERVICE *
C     *       DATE: 15 APRIL 76                                      *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                8120 WOODMONT AVENUE                          *
C     *                BETHESDA,MARYLAND 20014                       *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
      INTEGER ARCID
C     ************        **********         ***********     ******
C
C                        *****FORMAT STATEMENTS*****
C
10    FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J1,J2,A4,2X,
     1F12.9)
20    FORMAT (4X,'Y',J5,4X,'RES',J5,2X,F12.9,3X,A1,J1,J2,A4,2X,F12.9)
30    FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.9)
40    FORMAT (4X,'XN',J2,6X,'N',J1,J2,A4,2X,F12.9,3X,'OBJECTIV',2X,
     1F12.9)
121   FORMAT('Y',J5,1X,4I1,J5,I4)
C
      IF (JGRADE.EQ.6) RETURN
      IF (ICHG.GT.NYRS) GO TO 999
      IF (JGRADE.EQ.2) GO TO 70
C        ***FOR Y MMNN, WHERE M=N,STARTING FROM T-1 SINCE NO INCUMBENTS
C                   FROM T=0 --- T-1**********
         DO 60 J=2,NYRS
         K=J-1
            DO 50 M=1,NSPEC
            CALL IPHASE(M,$50)
            ARCID=(K*10000)+(NBRSPC(M)*100)+NBRSPC(M)
C        ************GOZO*******           ******GOZO******
            WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,NBRSPC(M)
     1      ,NAME(3),SURVHI(J)
C        *********CINC*****                ****CREQ*******
            WRITE (9,10) ARCID,K,NBRSPC(M),NAME(5),MINUS1,NN,J,NBRSPC(M)
     1      ,NAME(1),PLUS1
            IF (J.EQ.NYRS) GO TO 50
C        *********TREQ********
            WRITE (9,10) ARCID,J,NBRSPC(M),NAME(2),PLUS1
50          CONTINUE
60       CONTINUE
```

```
C         **FOR Y=M+N,WHERE M.NE.N,FROM Y=0 FOR GRADES 4&5
C                                          FROM ICHG FOR GRADE 2
70      IBGN=1
        IF (JGRADE.EQ.2) IBGN=ICHG+1
        IF (IBGN .GT. NYRS) GO TO 999
           DO 270 J=IBGN,NYRS
           K=J-1
              DO 260 M=1,NSPEC
              CALL IPHASE(M,$260)
                 DO 250 N=1,NSPEC
                 CALL IPHASE(N,$250)
                 CALL JPHASE(M,N,$250)
                 FLOFAC=1.0
C          *** USED IN COEFF CALCULATION -  IS 1.0 UNLESS SOURCE FOR
C             HIGHER GRADE FED FROM BOTH PRIMARY AND ALTERNATE SPEC,
C             IN WHICH CASE FLOFAC =0.5  **********
                 IF (M.EQ.N) GO TO 250
                 INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
                 IF (IBITS(RESFLO,INDEX).EQ.0) GO TO 250
                 ARCID=K*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
C           *******GOZO****                    ****GOZO****
                 WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,
     1           NBRSPC(N),NAME(3),SURVMI(J)
                 IBIT=0
                 IF (JGRADE.EQ.2.AND.IBITS(RESFLO,INDEX).EQ.4) IBIT=2
                 IF (JGRADE.EQ.2.AND.IBITS(RESFLO,INDEX).NE.4) IBIT=1
C            *****RESFLO****                 ****CREQ*******
                 IF (JGRADE.GT.3) WRITE (9,20) ARCID,ARCID,MINUS1,NN,J,
     1           NBRSPC(N),NAME(1),PLUS1
C            *****RESFLO****                 ****CREQ******
                 IF (IBIT.EQ.2) WRITE (9,20) ARCID,ARCID,MINUS1,NN,J,
     1           NBRSPC(N),NAME(1),PLUS1
C                  *****CREQ****
                 IF (IBIT.EQ.1) WRITE (9,30) ARCID,J,NBRSPC(N),NAME(1),
     1           PLUS1
                 IF (J.EQ.NYRS) GO TO 250
C                       *******TREQ******
                 WRITE (9,10) ARCID,J,NBRSPC(N),NAME(2),PLUS1
C          DETERMINE IF FLOW IS FROM PRIMARY OR ALTERNATE OR BOTH SPEC
C          IF NONEOF THE ABOVE, THEN IS IN THE PATH OF AN EARLIER SOURCE
                 IF (IBITS(RESFLO,INDEX).EQ.1) GO TO 80
                 IF (IBITS(RESFLO,INDEX).EQ.2) GO TO 90
                 IF (IBITS(RESFLO,INDEX).NE.3) GO TO 250
                 FLOFAC=0.5
C         PROMOTION IS FROM PRIMARY,THEREFORE M IS THE PRI SPEC
80               MRATIO=YUTIL(M,N)/10
                 NRATIO=YUTIL(M,N)-(MRATIO*10)
                 GO TO 100
C        PROMOTION IS FROM THE ALTERNATE,THEREFORE M IS TH ALT SPEC
90               NRATIO=YUTIL(N,M)/10
                 MRATIO=YUTIL(N,M)-(NRATIO*10)
100              LENGTH=NRATIO*YTOUR(N)
                 KT=K
                 IF ((KT+LENGTH).GE.NYRS) GO TO 210
C        ****THERE WILL BE FLOW OUT OF SPEC N****
                 KPER=KT+LENGTH
C        ***HOW MANY CINC CONSTRAINTS IN YTOUR LENGTH
```

```
            HOLD1=1.0
110         IEND=LENGTH-1
            IF (IEND.GT.0) GO TO 120
C         ..ONE YEAR YTOUR...
            N1TOUR=N1TOUR+1
            IF (N1TOUR.GT.1) GO TO 200
            KPER2=K
            GO TO 150
120            DO 130 N1=1,IEND,2
               KPER1=K+N1
               KPER2=KPER1+1
               VALUE1=HOLD1*SURVH1(KPER1)*FLOFAC
               VALUE2=(VALUE1*SURVH1(KPER2))*FLOFAC
               IF ((N1+1).GT.IEND) GO TO 140
C              ....CINC..                    ..CINC....
               WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1,NN,
     1         KPER2,NBRSPC(N),NAME(5),VALUE2
               HOLD1=VALUE2
130            CONTINUE
            GO TO 150
C              ..CINC...
140         WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(5),VALUE1
            HOLD1=VALUE1
            KPER2=KPER1
C        IS THERE TIME TO ROTATE BACK TO SPEC M
150         IF ((KPER2+1).GE.NYRS) GO TO 240
C         ...RESTRICTIVE FLOW FOR KPER N TO M.......
            IDOUT=(KPER*10000)+NBRSPC(N)*100+NBRSPC(M)
            HOLD1=HOLD1*SURVH1(KPER)
C          .....RESFLO.....
            WRITE (9,20) ARCID,IDOUT,HOLD1
      IVAL1 = HOLD1*1000
      WRITE(8,121)IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),
     * IDOUT,IVAL1
C     ROTATE TO M AND DETERMINE TIME IN SPEC M
            LENGTH=MRATIO*YTOUR(M)
            IF ((KPER+LENGTH-1).GE.NYRS) LENGTH=NYRS-KPER
C      ...FULL YTOUR
C     ....HOW MANY CINC CONSTRAINTS FOR SPEC M
            IEND=LENGTH-1
            IF (IEND.GT.0) GO TO 160
C         ...ONE YEAR YTOUR..
            KPER2=KPER2+1
            GO TO 190
160            DO 170 M1=1,IEND,2
               KPER1=KPER+M1
               KPER2=KPER1+1
               VALUE1=HOLD1*SURVH1(KPER1)*FLOFAC
               VALUE2=VALUE1*SURVH1(KPER2)*FLOFAC
               IF ((M1+1).GT.IEND) GO TO 180
C              ...CINC..                 ..CINC..
               WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(5),VALUE1,NN,
     1         KPER2,NBRSPC(M),NAME(5),VALUE2
               HOLD1=VALUE2
170            CONTINUE
            GO TO 190
C                    ...CINC...
```

```
180         WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(S),VALUE1
            HOLD1=VALUE1
            KPER2=KPER1
C     ***ARE THERE ANY PERIODS BEYOND KPER2
190         IF ((KPER2+1).GE.NYRS) GO TO 240
C    ****** ROTATE TO SPEC N *******
            K=KPER2
C       **ADD THE LENGTH OF M***
            KPER=KPER+LENGTH
            IF (KPER.GE.NYRS) GO TO 240
            IDOUT=(KPER)*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
            HOLD1=HOLD1*SURVHI(KPER)*FLOFAC
            VALUE1=HOLD1
C     ***** WRITE OUT RESTRICTIVE FLOW FOR PERIOD KPER  M TO N  ***
C                    ****RES--M-N******
            WRITE (9,20) ARCID,IDOUT,VALUE1
      IVAL1 = VALUE1*1000
      WRITE(8,121)IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),
     * IDOUT,IVAL1
            LENGTH=YTOUR(N)*NRATIO
C          ***ADD THE LENGT OF N***
            KPER=KPER+LENGTH
C     **IS THERE TIME TO STAY IN N FOR 1 MORE YR**
            K=K+1
            IF (K.LT.NYRS) GO TO 110
            GO TO 240
C      **SPEC N IS ONE YEAR LONG***
200         IDOUT=(KPER*10000)+(NBRSPC(N)*100)+NBRSPC(M)
            HOLD1=HOLD1*SURVHI(KPER)*FLOFAC
            VALUE1=HOLD1
C                  ***RES--N-M****
            WRITE (9,20) ARCID,IDOUT,VALUE1
      IVAL1 = VALUE1*1000
      WRITE(8,121)IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),
     * IDOUT,IVAL1
            KPER2=KPER
            GO TO 190
C       **TIME IN SPEC N EXCEEDS HORIZON-WILL NOT LEAVE N***
210         IF (K.EQ.(NYRS-1)) GO TO 250
C                             **CINC***
            WRITE (9,10) ARCID,J,NBRSPC(N),NAME(S),SURVHI(J)
C          **HOW MANY MORE CINCS***
            IEND=NYRS-(K+2)
            IF (IEND.LE.0) GO TO 250
            RLEFT=SURVHI(J)
               DO 230 L=1,IEND,2
               KPER1=K+L+1
               KPER2=KPER1+1
               VALUE1=RLEFT*SURVHI(KPER1)
               VALUE2=VALUE1*SURVHI(KPER2)
               IF ((L+1).GT.IEND) GO TO 220
C                ***CINC**               ***CINC***
               WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(S),VALUE1,NN,
     1         KPER2,NBRSPC(N),NAME(S),VALUE2
               RLEFT=VALUE2
               GO TO 230
C             ***CINC***
```

```
220               WRITE (9,10) ARCID,KPERI,NBRSPC(N),NAME(5),VALUE1
230               CONTINUE
               GO TO 250
240            K=KT
               INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               IF (IBITS(RESFLO,INDEX).NE.3) GO TO 250
               GO TO 90
250            CONTINUE
260         CONTINUE
270      CONTINUE
      IF (JGRADE.NE.2) RETURN
C           *****WRITE CONSTRAINTS FOR XN--****
999      DO 280 J=1,NSPEC
         CALL IPHASE(J,$280)
C         ******N MMGOZO****          ****OBJECTIV****
         WRITE (9,40) NBRSPC(J),NYRS,NBRSPC(J),NAME(3),MINUS1,PLUS1
280      CONTINUE
C        ********              *******          *********
      RETURN
      END
```

```
      FUNCTION IBITS (IWORD,JBIT)
C     ******************************************************************
C     ******************************************************************
C     *                                                                *
C     *          FUNCTION:   IBITS                                     *
C     *          CALLED BY:  BOUNDS    CPDIAG    HICOLS    LOCOLC      *
C     *                      LOCOLS    LODIAG    RESHI     RESLO       *
C     *                      ROWOP                                     *
C     *          CALLING  ARGUMENTS:                                   *
C     *                               IWORD -  ARRAY TO BE PACKED      *
C     *                               JBIT  -  INDEX OF THE PACKED ARRAY *
C     *                               JVALUE-  VALUE TO BE STORED IN ARRAY *
C     *          CALLED ROUTINES: NONE                                 *
C     *          PURPOSE:  THIS FUNCTION/SUBROUTINE HANDLES THE        *
C     *                    PACKING AND UNPACKING FOR THE MATRIX        *
C     *                    GENERATOR. EACH VALUE OF A WORD IS          *
C     *                    PACKED INTO 3 BITS THUS REDUCING CORE       *
C     *                    REQUIREMENTS FOR THE ARRAY BY 11/12THS.     *
C     *                    THESE WORDS INDICATE WHETHER A RESTRICTIVE  *
C     *                    FLOW CONTRAINT IS REQUIRED.                 *
C     *          DATE: 15 APRIL 76                                     *
C     *          AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *          COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                   8120 WOODMONT AVENUE                         *
C     *                   BETHESDA,MARYLAND 20014                      *
C     *                                                                *
C     ******************************************************************
C
C*********    THE ENTRY POINT IBITS RECALLS THE STORED VALUE
C
      DIMENSION IWORD(1)
      WWW(1)=IWORD(1)
      IBITS=FLD(3*MOD(JBIT+11,12),3,WWW((JBIT+11)/12))
      RETURN
C
C*********    THE ENTRY POINT SET STORES THE VALUE
      ENTRY SET(IWORD,JBIT,JVALUE)
      FLD(3*MOD(JBIT+11,12),3,WWW((JBIT+11)/12))=JVALUE
      RETURN
      END
```

```
      SUBROUTINE INPUT
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *      SUBROUTINE: INPUT                                       *
C     *      CALLED BY: MAIN                                         *
C     *      CALLING  ARGUMENTS: NONE                                *
C     *      CALLED ROUTINES: IPHASE,R8EDIT(INTERNAL)                *
C     *      INPUT FILES:                                            *
C     *                      5 - ODINPUD01.                          *
C     *                          CARD 1 - BASIC PARAMETERS           *
C     *                          CARD 2 - AUTHORIZED STRENGTH/GRADE  *
C     *                          CARD 3 - SUBSET 1 SPECIALTIES       *
C     *                          CARD 4 - ATTRITION FACTOR - PROMOTEES *
C     *                          CARD 5 - ATTRITION FACTOR - IN GRADE *
C     *                          CARD 6 - PROMOTION FACTOR           *
C     *                          CARD 7 - OVERFLOW/UNDERFLOW FACTORS *
C     *                          CARD 8 - SEGMENTATION INDICATOR     *
C     *                          CARD 9 - U.B. FOR SUBSET 1          *
C     *                          CARD 10- U.B. FOR ALT SPEC SUBSET 1 *
C     *                          CARD 11- ADV ENTRY SPEC,ICHG,NBRAES *
C     *                          CARD 12-  % OF CPT REMAINING/YEAR   *
C     *                     10 - ODRATUD01.                          *
C     *                          UTILIZATION  RATIOS  AND            *
C     *                          TOUR LENGTHS FOR THE GRADE          *
C     *                     11 - ODSACUD01.                          *
C     *                          SPECIALTY REQUIREMENTS/GRADE        *
C     *      OUTPUT FILES:                                           *
C     *                      6 - STANDARD OUTPUT ON PRINTER          *
C     *                      9 - ODEQAUD01.                          *
C     *      PURPOSE: THIS ROUTINE INPUTS ALL THE DATA INTO          *
C     *               THE MATRIX GENERATOR.IT ALSO COMPUTES          *
C     *               THE TOTAL REQUIREMENTS FOR A GRADE BY ADDING   *
C     *               THE UNFILLED HIGHER GRADE REQUIREMENTS TO THE  *
C     *               REQUIREMENTS OF THE GRADE SEGMENT. ADDITIONALLY *
C     *               RECOMPUTES PROMOTION RATES FOR LTC AND MAJ     *
C     *               SEGMENTS BASED UPON SOLUTION TO PREVIOUS       *
C     *               SEGMENT.                                       *
C     *      DATE: 15 APRIL 76                                       *
C     *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R. L. BROWN *
C     *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *               8120 WOODMONT AVENUE                           *
C     *               BETHESDA,MARYLAND 20014                        *
C     *                                                              *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1,LIST
      CALL OPT(IVALUE)
      DIMENSION ITREQ(11,6,100),ATTHI(9),ATTLO(9)
      DATA SARRAY/50*1./
      DEFINE FILE10(12,2500,U,IREC)
C
C                     *****FORMAT STATEMENTS*****
C
    1 FORMAT (I2,2X,I2,3X,2A4,2X,A4,I1)
```

```
    2     FORMAT ('NAME',10X,2A4)
    3     FORMAT ('ROWS'/1X,'N  OBJECTIV')
    4     FORMAT ()
    5     FORMAT (1H1,48X,'UNFILLED REQUIREMENTS FOR GRADE O-',I1,/)
    6     FORMAT (10X,'           PCT AUTH        TO',8X,9('T',I1,8X))
    7     FORMAT (1H0,'SPECIALTY',I3,F10.2,10I10)
    8     FORMAT (1X,22I3)
    9     FORMAT (1X,9F8.5)
   10     FORMAT (1H1,53X,'REQUIREMENTS FOR GRADE O-',I1,/)
   11     FORMAT (10F6.0)
   12     FORMAT (10F6.3)
   13     FORMAT (' OFFICER DUAL SPECIALTY ALLOCATION RUN FOR GRADE',I3)
   14     FORMAT (' SEGMENT CODE ',I3,'  (0=NO SEGMENTATION  1-FIRST',
         * ' SEGMENT  2-SECOND SEGMENT')
C
C
          DO 20 N = 1,50
   20     SARRAY(N) = 1.0
C
C         READ IN PARAMETERS
          READ (5,  1) NSPEC,NYRS,NAME1,NAME2,MODE,JGRADE
          IF (NSPEC . GT. MAXSPC) STOP NSPEC
          IF (NYRS  . GT. MAXYRS) STOP NYRS
C
          WRITE (9,  2) NAME1,NAME2
          WRITE (9,  3)
          READ(5,4)AUTHMX
C
C         READ IN SELECTED SPECIALITIES FOR SEGMENT 1
          READ (5,  4) NBRPRO,(NPROB(K),K=1,NBRPRO)
C
C         READ IN ATTRITION AND PROMOTION FACTORS
C
C         READ IN ATTHI
          READ (5,  4) (ATTHI(K),K=1,NYRS)
C
C         READ IN ATTLO
          READ (5,  4) (ATTLO(K),K=1,NYRS)
C
C         READ IN PRMT
          READ (5,  4) (PRMT(K),K=1,NYRS)
C
              DO 200 J=1,NYRS
              SURVHI(J)=1.-ATTHI(J)
              SURVLO(J)=1.-ATTLO(J)
  200         CONTINUE
C
C         READ IN OVERFLOW AND UNDERFLOW VARIABLES
          READ (5,  4) OFLOLO,UFLOLO,OFLOHI,UFLOHI
C
          WRITE(6,13)JGRADE
          IF (JGRADE.LE.3) GO TO 205
          READ(5,4) ISEG                          ! SEGMENT INDICATOR (0,1,2)
          WRITE(6,14)ISEG
C
          READ(5,11)UPBND
          READ(5,12)SARRAY
```

```
      GO TO 210
C
C ****** READ IN ADVANCED ENTRY SPECIALTIES,ICHG,NBR OF ADVANCED
C ****** SPECIALTIES
205   READ (5,  8) (AES(IN),IN=1,20),ICHG,NBRAES
C
C*********READ IN NUMBER OF OFFICERS REMAINING***********
      READ (5,  9) CPTREM
C     READ IN UTILIZATION RATIOS FOR THE GRADE
210   IREC=7-JGRADE
      IF (JGRADE.GT.3) READ (10'IREC) UTIL
      IF (JGRADE.EQ.6) GO TO 220
      IREC=7-JGRADE
      IF (JGRADE.LE.3) IREC=3
      READ (10'IREC) YUTIL
C
C     READ IN TOUR LENGTHS FOR THE GRADE
220   IREC=10-JGRADE
      IF (JGRADE.GT.3) READ (10'IREC) TOUR
      IF (JGRADE.EQ.6) GO TO 230
      IREC=10-JGRADE
      IF (JGRADE.LE.3) IREC=6
      READ (10'IREC) YTOUR
C
C     READ IN VALUES FOR GRADE REQUIREMENTS
C     ITREQ(YR,JGRADE,SPEC)
C     READ IN REQUIREMENTS
230   READ (11) ITREQ
      IEND=NYRS+1
         DO 260 M=1,NSPEC
         I=NBRSPC(M)
            DO 250 K=1,IEND
            INDEX=(M-1)*IEND+K
            IGRADE=JGRADE+1
            IF (IGRADE.GT.6) GO TO 240
C
C     REQ1 HOLDS  THE HIGHER GRADE  REQUIREMENTS
            REQ1(INDEX)=FLOAT(ITREQ(K,IGRADE,I))*(1.+OFLOHI)+.5
240         CONTINUE
C     REQ2 HOLDS  THE CURRENT GRADE  REQUIREMENTS
            REQ2(INDEX)=FLOAT(ITREQ(K,JGRADE,I))*(1.+OFLOLO)+.5
250         CONTINUE
260      CONTINUE
C     **EDIT PROMOTION RATES FOR LTC AND MAJOR SEGMENTS
      IF(JGRADE.GT.3 .AND. JGRADE.LT.6)CALL RBEDIT
      IF (IGRADE.GT.6) GO TO 280
      IBGN=1
      IEND=NYRS+1
      WRITE (6,  5) IGRADE
      WRITE (6,  6) (NL,NL=1,9)
         DO 270 M=1,NSPEC
      CALL IPHASE(M,$275)
         YARRAY = SARRAY(M)*100.
         WRITE (6,  7) NBRSPC(M),YARRAY,(REQ1(J),J=IBGN,IEND)
275      IBGN=IEND+1
         IEND=IEND+(NYRS+1)
270      CONTINUE
```

```
280     IBGN=1
        IEND=NYRS+1
        WRITE (6, 10) JGRADE
        WRITE (6,  6) (NL,NL=1,9)
           DO 290 M=1,NSPEC
             YARRAY = SARRAY(M)*100.
        CALL IPHASE(M,$295)
           WRITE (6,  7) NBRSPC(M),YARRAY,(REQ2(J),J=IBGN,IEND)
295        IBGN=IEND+1
           IEND=IEND+(NYRS+1)
290        CONTINUE
C
C       COMPUTE TOTAL REQUIREMENTS AND STORE IN REQ2 ARRAY
        IF (JGRADE.EQ.6) RETURN
        IEND=NSPEC*(NYRS+1)
           DO 300 J=1,IEND
           REQ2(J)=REQ2(J)+REQ1(J)
300        CONTINUE
        RETURN
C
              SUBROUTINE RBEDIT
              REAL NUM
13      FORMA((1HO,20X,'***DERIVATION OF PROMOTION RATE***')
14            FORMAT(1HO,'**INITIALIZATION PHASE PROMOTION RATE FOR YEAR',
       *I3,'  RECOMPUTED BASED UPON PREVIOUS SEGMENT '/1HO,
       *'OLD RATE WAS 'F9.5,' VS NEW RATE ',F9.5/
       *' THEREFORE ',F9.5,' WILL BE USED IN THIS SEGMENT')
15      FORMAT(1HO,'TOTAL UNFILLED REQUIREMENTS',34X,F12.3)
16      FORMAT(1HO,'MINUS: REQUIREMENTS SATISIFIED LAST YEAR',21X,
       *F12.3)
17      FORMAT(1HO,5X,'RAW REQUIREMENTS TO BE FILLED THIS YEAR',17X,
       *F12.3)
18      FORMAT(1HO,'PLUS: ATTRITION IN GRADE',I2,
       *' LAST YEAR(ATTRITION RATE =',F6.4,') ',F12.3)
19      FORMAT(1HO,5X,'TOTAL REQUIREMENTS TO BE FILLED BY',
       *' PROMOTION THIS YEAR ',1X,F12.3,//)
21      FORMAT(1HO,'STARTING GRADE',I2,' POPULATION LAST YEAR',24X,
       *F12.3)
22      FORMAT(1HO,'MINUS: PROMOTIONS LAST YEAR',34X,F12.3)
23      FORMAT(1HO,4X,'NON-PROMOTED GRADE',I2,' POPULATION',26X,
       *F12.3)
24      FORMAT(1HO,'MINUS: ATTRITION OF NON-PROMOTED(ATTRITION RATE = ',
       *F6.4,')',4X,F12.3)
25      FORMAT(1HO,'GRADE',I2,' POPULATION AVAILABLE FOR PROMOTION',19X,
       *F12.3,//)
26      FORMAT(1HO,10X,'PROMOTION RATE = TOTAL REQUIREMENTS TO BE FILLED ',
       *' BY PROMOTION /')
27      FORMAT(1HO,30X,'GRADE',I2,' POPULATION AVAILABLE FOR PROMOTION')
28      FORMAT(1HO,24X,' = ',F12.3,' / ',F12.3)
29      FORMAT(1HO,24X,' = ',F12.3)
              TAUTH=AUTHMX(JGRADE)
              SAVHI=0.0
              SAVDIF=0.0
              SAVSUM=0.0
              IEND=NYRS+1
              DO 500 L=2,IEND
         SUM=0.0
```

```
          DO 400 K=1,NSPEC
      KK=NBRSPC(K)
          SUM=SUM+FLOAT(ITREQ(L,IGRADE,KK)*(1.0+OFLOHI))
400       CONTINUE
          ATTRHI=0.0
          IF(L.GT.2)ATTRHI=(SAVSUM)*ATTHI(L-2)
          UFREQ=SUM-SAVSUM
          NUM=UFREQ+ATTRHI
          ATTRLO=0.0
          IF(L.GT.2)ATTRLO=(TAUTH-SAVCIF-SAVHI)*ATTLO(L-2)
          DENOM=TAUTH-(SAVDIF+SAVHI)-ATTRLO
          IN=L-1
          R8=NUM/DENOM
      REQ4=SAVHI+SAVDIF
      REQ3=TAUTH-REQ4
          WRITE(6,14)IN,PRMT(IN),R8,R8
      WRITE(6,13)
      WRITE(6,15)SUM
      WRITE(6,16)SAVSUM
      WRITE(6,17)UFREQ
      WRITE(6,18)IGRADE,ATTHI(L-2),ATTRHI
      WRITE(6,19)NUM
      WRITE(6,21)JGRADE,TAUTH
      WRITE(6,22)REQ4
      WRITE(6,23)JGRADE,REQ3
      WRITE(6,24)ATTLO(L-2),ATTRLO
      WRITE(6,25)JGRADE,DENOM
      WRITE(6,26)
      WRITE(6,27)JGRADE
      WRITE(6,28)NUM,DENOM
      WRITE(6,29)R8
          PRMT(IN)=R8
          TAUTH=DENOM
          SAVHI=ATTRHI
          SAVDIF=SUM-SAVSUM
          SAVSUM=SUM
500       CONTINUE
          RETURN
      END
```

```
      SUBROUTINE IPHASE(M,$)
C     ***************************************************************
C     ***************************************************************
C     *                                                             *
C     *     SUBROUTINE: IPHASE                                      *
C     *     CALLED BY: BOUNDS                                       *
C     *     CALLING  ARGUMENTS:                                     *
C     *                         M - SPECIALITY INDEX                *
C     *                         $ - RETURN IF M IS SUBSET 1 SPEC    *
C     *     CALLED FUNCTION: IPROB                                  *
C     *     PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO DETERMINE    *
C     *              IF THE SEGMENT INDICATOR IS SET AT TWO.        *
C     *               A NONSTANDARD RETURN RESULTS IF M IS A        *
C     *              SUBSET 1 SPECIALTY,OTHERWISE RETURN            *
C     *     DATE: 15 APRIL 76                                       *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *              8120 WOODMONT AVENUE                           *
C     *              BETHESDA,MARYLAND 20014                        *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1
      IF (ISEG .LT. 2) RETURN
      IF (IPROB(M).EQ.1) RETURN 2
      RETURN
      END
```

```
      FUNCTION   IPROB(M)
C     ***************************************************************
C     *                                                             *
C     *     FUNCTION: IPROB                                         *
C     *     CALLED BY: IPHASE,JPHASE                                *
C     *     CALLING ARGUMENTS:                                      *
C     *                     : M - SPECIALTY INDEX                   *
C     *     PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO DETERMINE    *
C     *              IF THE SPECIALTY RECEIVED AS AN ARGUMENT       *
C     *              IS CONTAINED IN THE LIST OF SUBSET 1 SPECIALTIES *
C     *     DATE: 15 APRIL 76                                       *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *              8120 WOODMONT AVENUE                           *
C     *              BETHESDA,MARYLAND 20014                        *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1
      IPROB = 0
      DO 10 NX = 1,NBRPRO
      IF (NBRSPC(M).EQ.NPROB(NX)) IPROB= 1
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE JPHASE(M,N,$)
C     ****************************************************************
C     *                                                              *
C     *     SUBROUTINE: JPHASE                                       *
C     *     CALLED BY: BOUNDS                                        *
C     *     CALLING ARGUMENTS:                                       *
C     *                          M - SPECIALTY INDEX                 *
C     *                          N - SPECIALTY INDEX                 *
C     *                          $ - RETURN IF M IS ALTERNATE SPEC   *
C     *     CALLED FUNCTION: IPROB                                   *
C     *     PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO DETERMINE IF  *
C     *              THE SEGMENT INDICATOR IS SET AT ONE.IF IT IS    *
C     *              1 A NONSTANDARD RETURN RESULTS IF EITHER M OR N *
C     *              IS NOT A SUBSET 1 SPECIALTY                     *
C     *     DATE: 15 APRIL 76                                        *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *              8120 WOODMONT AVENUE                            *
C     *              BETHESDA,MARYLAND 20014                         *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
      IF (ISEG .NE. 1) RETURN
      IF (IPROB(M)*IPROB(N) .LT. 1) RETURN 3
      RETURN
      END
```

```
      SUBROUTINE KEYARC
C     ******************************************************************
C     *                                                                *
C     *       SUBROUTINE: KEYARC                                       *
C     *       CALLED BY: MAIN                                          *
C     *       CALLING ARGUMENTS: NONE                                  *
C     *       CALLED ROUTINE: IPHASE,JPHASE                            *
C     *       OUTPUT FILE: - 9 -  ODEQAUD01.                           *
C     *       PURPOSE: NAMES THE KEY ARC RELATIONSHIP CONSTRAINTS.     *
C     *       DATE: 15 APRIL 76                                        *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *                8120 WOODMONT AVENUE                            *
C     *                BETHESDA,MARYLAND 20014                         *
C     *                                                                *
C     ******************************************************************
C
C
      INCLUDE PROC1
C
C   ****** FORMAT STATEMENTS
C
10    FORMAT (1X,'L  UR',J2,J2)
      IF(JGRADE.LT.3) RETURN
C      ***NAME KEYARC CONST.,UR=M=N,WHERE M .LT. N
      NSPECX = NSPEC - 1
      DO 100  M = 1,NSPECX
      CALL IPHASE(M,$100)
      MM = M + 1
      DO 101  N = MM,NSPEC
      CALL IPHASE(N,$101)       @ SEGMENT CHECK
      CALL JPHASE(M,N,$101)
         IF(JGRADE .GT. 3) GO TO 50
      IF(YUTIL(M,N) .LT.88) WRITE(9,10)NBRSPC(M),NBRSPC(N)
      IF(YUTIL(M,N).EQ.88.AND.YUTIL(N,M).LT.88) WRITE(9,10)
     * NBRSPC(M),NBRSPC(N)
         GO TO 101
50     IF(UTIL(M,N) .LT.88) WRITE(9,10)NBRSPC(M),NBRSPC(N)
      IF(UTIL(M,N).EQ.88.AND.UTIL(N,M).LT.88) WRITE(9,10)
     * NBRSPC(M),NBRSPC(N)
101   CONTINUE
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOCOL
C     ***************************************************************
C     *                                                             *
C     *       SUBROUTINE: LOCOL                                     *
C     *       CALLED BY: MAIN                                       *
C     *       CALLING  ARGUMENTS: NONE                              *
C     *       CALLED ROUTINES: LOCOLC,LOCOLL,LOCOLS,IPHASE,JPHASE   *
C     *       PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO DETERMINE  *
C     *                WHICH LOCOL SUBROUTINE TO CALL BASED UPON    *
C     *                THE VALUE OF JGRADE,AND TO WRITE STATISTICAL *
C     *                SUMMARY OF MATRIX GENERATOR ACTIVITY. THE    *
      *                NUMBER OF PREFERRED SPECIALTY PAIRS          *
      *                (NPREF) IS ALSO COMPUTED AND DISPLAYED.      *
C     *                THE NUMBER OF PREFERRED SPECIALTY PAIRS      *
C     *                (NPREF) IS ALSO COMPUTED AND DISPLAYED.      *
C     *       DATE: 15 APRIL 76                                     *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                8120 WOODMONT AVENUE                         *
C     *                BETHESDA,MARYLAND 20014                      *
C     *                                                             *
C     ***************************************************************
      INCLUDE PHOC1
C******* WRITE ALL CONSTRAINT INFORMATION   *******
C
C
      IF (JGRADE .GT. 3) GO TO 50
      DO 90 M=1,NSPEC
      DO 80 N=1,NSPEC
      IF (YUTIL(M,N) .LT. 88) NPREF=NPREF+1
      IF (YUTIL(M,N) .EQ. 88 .AND. YUTIL(N,M) .LT. 88)NPREF=NPREF+1
80    CONTINUE
90    CONTINUE
      GO TO 200
50    DO 100  M = 1,NSPEC
      CALL IPHASE(M,$100)
      DO 101  N = 1,NSPEC
      CALL IPHASE(N,$101)
      CALL JPHASE(M,N,$101)
      IF(UTIL(M,N) .LT.88) NPREF = NPREF + 1
      IF(UTIL(M,N).EQ.88.AND.UTIL(N,M).LT.88) NPREF = NPREF + 1
101   CONTINUE
100   CONTINUE
200   I1=1
      ITOT2 = IROWCT(1) + IROWCT(2)
      ITOT3 = IROWCT(4) + IROWCT(6)
      ITOT4 = IROWCT(5) + IROWCT(7)
      ITOT1 = IROWCT(3)
      IF (ISEG.LE.1.AND.JGRADE.GT.3 )ITOT5 = NBRPRO
      IF (ISEG.NE.1) ITOT5 = ITOT5 + 1
      IT = NPREF/2
        IF(JGRADE .LT. 3) IT = 0
      ITOT6 = ITOT1+ITOT2+ITOT3+ITOT4+ITOT5 + I1 + IT
      WRITE(6,1)ISEG,JGRADE
1     FORMAT('1SEGMENT NUMBER **',I1,'** FOR GRADE EQUAL TO **',I1,'**')
```

```
      WRITE(6,2)II,ITOT1,ITOT2,IROWCT(I),ITOT3,ITOT4,ITOT5,IT,ITOT6
2     FORMAT(1H0,//////,20X,'CONSTRAINTS BY TYPE    ',/,
     * 15X, ' OBJECTIVE FUNCTION                 ',I5,/
     * 15X, ' FLOW CONSERVATION                  ',I5,/
     * 15X, ' REQUIREMENTS - TOTAL               ',I5,
     *  '   (HIGHER GRADE =     ',I5,')',/
     * 15X, ' CONTROL OF X-ARCS                  ',I5,/,
     * 15X, ' CONTROL OF Y-ARCS(PROMOTEES) ',I5,/,
     * 15X, ' CONTROL OF INPUT                   ',I5,/,
     * 15X, ' KEY ARC RELATIONSHIPS              ',I5,/,
     * 15X, '      TOTAL CONSTRAINTS             ',I5,/////)
      JSPEC = NSPEC
      IF (ISEG.EQ.2) JSPEC = NSPEC - NBRPRO
      WRITE (6,903) JSPEC,NYRS,NPREF
903   FORMAT(15X,' NUMBER OF SPECIALTIES EQUAL  ',I5,/,
     *      15X,' NUMBER OF YEARS EQUAL          ',I5,/,
     *      15X,' NUMBER OF PREFERENCES EQUAL  ',I5,/,1H1)
      IF (JGRADE.GT.3) CALL LOCOLS
      IF (JGRADE.EQ.3) CALL LOCOLC
      IF (JGRADE.LT.3) CALL LOCOLL
      RETURN
      END
```

```
      SUBROUTINE LOCOLC
C     ****************************************************************
C     *                                                              *
C     *       SUBROUTINE: LOCOLC                                     *
C     *       CALLED BY: LOCOL                                       *
C     *       CALLING  ARGUMENTS: NONE                               *
C     *       CALLED ROUTINES: VALID                                 *
C     *       OUTPUT FILES:                                          *
C     *                      8 - ODSAPUD08.                          *
C     *                      9 - ODEQAUD01.                          *
C     *       PURPOSE:  DEFINES XOOOM, WOM-M, WOM-N, X-M-M AND       *
C     *                 Y-M-M ARCS FOR CPT SEGMENT.                  *
C     *       DATE: 15 APRIL 78                                      *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                8120 WOODMONT AVENUE                          *
C     *                BETHESDA,MARYLAND 20014                       *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
      INTEGER ARCID
C
C                         *****FORMAT STATEMENTS*****
C
10    FORMAT ('COLUMNS')
20    FORMAT (4X,'W',J5,4X,'WO',J2,A4,2X,F12.9,3X,A1,J3,A4,2X,F12.9)
30    FORMAT (4X,'W',J5,4X,'N',J3,A4,2X,F12.9,3X,A3,J5,2X,F12.9)
40    FORMAT (4X,'W',J5,4X,'RES',J5,2X,F12.9,3X,A1,J3,A4,2X,F12.9)
45    FORMAT (4X,'W',J5,4X,'TOTAUTH ',2X,F12.9)
46    FORMAT (4X,'W',J5,4X,'UR',2J2,4X,F12.9)
50    FORMAT (4X,'W',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A3,J5,4X,F12.9)
60    FORMAT (4X,'XOOO',J2,4X,'WO',J2,A4,2X,F12.9)
70    FORMAT (4X,'XOOO',J2,4X,'NO',J2,A4,2X,F12.9,3X,'WO',J2,A4,2X,
     1F12.9)
80    FORMAT (4X,'XOOO',J2,4X,'N',J1,J2,A4,2X,F12.9)
90    FORMAT (4X,'W',J5,4X,'WO',J2,A4,2X,F12.9,3X,'NO',J2,A4,2X,F12.9)
100   FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,'N',J1,J2,A4,2X,
     1F12.9)
110   FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9)
120   FORMAT (4X,'W',J5,4X,A3,J5,2X,F12.9,3X,A3,J5,2X,F12.9)
130   FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,'N',J1,J2,A4,2X,
     1F12.9)
140   FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.9)
145   FORMAT('Y',J5,1X,4I1,9X,J5,J4)
C
C
C     ****** GENERATE COLUMNS  CHAPTER*******
C         ********          *******          *******
      WRITE (9,10)
C          ****COMPUTING COEFF FOR INITIAL FLOWS
C         **ENTRIES FOR XOOOMM**
         DO 160 M=1,NSPEC
C             **GOZO**
         WRITE (9,60) NBRSPC(M),NBRSPC(M),NAME(3),PLUS1
         IF (VALID(M).GT.0) GO TO 160
```

```
C             ***NO**LINC**                  **WO**LINC**
         WRITE (9,70) NBRSPC(M),NBRSPC(M),NAME(4),CPTREM(2),NBRSPC(M),
     1   NAME(4),CPTREM(1)
         IF (ICHG.LT.2) GO TO 160
         ILAST=ICHG-1
      IF (ICHG.GT.NYRS)ILAST = NYRS - 1
            DO 150 J=1,ILAST
            K=J+2
C          **NJ  LINC**
            WRITE (9,80) NBRSPC(M),J,NBRSPC(M),NAME(4),CPTREM(K)
150         CONTINUE
160      CONTINUE
C
C*****FOR INCUMBENT ARCS IN TO' TO TO
C
         DO 170 M=1,NSPEC
         IF (VALID(M).GT.0) GO TO 170
         ARCID=NBRSPC(M)*100+NBRSPC(M)
C                ******WO  GOZO**                **NO  GOZO**
         WRITE (9,90) ARCID,NBRSPC(M),NAME(3),MINUS1,NBRSPC(M),NAME(3),
     1   PLUS1
C            **WO LINC**                       **NO  TREQ**
         WRITE (9,90) ARCID,NBRSPC(M),NAME(4),MINUS1,NBRSPC(M),NAME(2),
     1   PLUS1
C        *** TOTAUTH ***
            WRITE (9,45) ARCID,PLUS1
170      CONTINUE
C******FOR INCUMBENT ARCS TO ---T,ICHG , CPT'S, LT 8 YEARS OF SERVICE
C
      ILAST = MIN(NYRS,ICHG)
         DO 190 J=1,ILAST
         K=J-1
            DO 180 M=1,NSPEC
            IF (VALID(M).GT.0) GO TO 180
            ARCID=(K*10000)+NBRSPC(M)*100+NBRSPC(M)
C           **NK  GOZO**               **NJ  GOZO**
            WRITE (9,100) ARCID,K,NBRSPC(M),NAME(3),MINUS1,J,NBRSPC(M),
     1      NAME(3),SURVLO(J)
C           **NJ  TREQ**               **NK  LINC**
            IF (J.NE.NYRS) WRITE (9,100) ARCID,J,NBRSPC(M),NAME(2),
     1      PLUS1,K,NBRSPC(M),NAME(4),MINUS1
C                                      **NK  LINC**
            IF (J.EQ.NYRS) WRITE (9,110) ARCID,K,NBRSPC(M),NAME(4),
     1      MINUS1
180         CONTINUE
190      CONTINUE
C
C******FOR INCUMBENT ARCS  CPT/MAJ W/GE  8 YEARS OF SERVICE
         DO 210 J=1,NYRS
         K=J-1
            DO 200 M=1,NSPEC
            ARCID=(K*10000)+NBRSPC(M)*100+NBRSPC(M)
C           **NK  GOZO**               **NJ  GOZO**
            WRITE (9,130) ARCID,K,NBRSPC(M),NAME(3),MINUS1,J,NBRSPC(M),
     1      NAME(3),SURVHI(J)
C           **NJ  TREQ**               **NK  CINC**
            IF (J.NE.NYRS) WRITE (9,130) ARCID,J,NBRSPC(M),NAME(2)
```

```
     1       PLUS1,K,NBRSPC(M),NAME(5),MINUS1
C                                     **NK  CINC**
             IF (J.EQ.NYRS) WRITE (9,140) ARCID,K,NBRSPC(M),NAME(5),
     1       MINUS1
200          CONTINUE
210       CONTINUE
C
C       T-ZERO PRIME VARIABLES  - WOMMNN
C
          DO 310 N=1,NSPEC
          UBKNT=0
             DO 300 M=1,NSPEC
C        **IF SHOULD NOT EXIST,COUNT TOWARDS UB ON NGSET LASTGO(N,M)=-1
             IF (YUTIL(M,N).EQ.99) GO TO 290
C          ***FOR FIELD GRADES M TO N ARCS DO NOT EXIST IN T-ZERO PRIME
             IF (M.EQ.N) GO TO 290
             IF ((YUTIL(M,N).EQ.88).AND.(YUTIL(N,M).GE.88)) GO TO 290
             IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
             ARCID=(NBRSPC(M)*100)+NBRSPC(N)
C      **GOZO**                  **GOZO**
             WRITE (9,20) ARCID,NBRSPC(M),NAME(3),MINUS1,NN,NBRSPC(N),
     1       NAME(3),PLUS1
C                 ****TREQ****
             WRITE (9,30) ARCID,NBRSPC(N),NAME(2),PLUS1
C            ****TOTAUTM*******
        WRITE(9,45) ARCID,PLUS1
C        **CHECK IF COMBAT ARMS SPECIALTY(88) IS INVOLVED
             IF (YUTIL(M,N).NE.88) GO TO 270
             IF (YUTIL(N,M).GE.88) GO TO 290
             NRATIO=YUTIL(N,M)/10
             MRATIO=YUTIL(N,M)-NRATIO*10
             LASTGO(N,M)=NRATIO*YTOUR(N)-1
             IF (LASTGO(N,M).GE.NYRS) LASTGO(N,M)=NYRS
             REMAIN=1.
             VALUE=REMAIN*(1.-(1./(YTOUR(N)*NRATIO)))
C      **VALUE = 0 IF YTOUR(N)*NRATIO =1, ALL LEAVE AT T-ZERO
220    CONTINUE
       I1 = MIN (M,N)
       I1 = NBRSPC(I1)
       I2 = MAX (M,N)
       I2 = NBRSPC(I2)
C
       RATE= MRATIO
       IF (N.GT.M) RATE = - RATE
C         *** UR-----CONSTRAINT ***
       WRITE(9,46)ARCID,I1,I2,RATE
             IF (VALUE.NE.0.) GO TO 230
       IREMX = REMAIN*1000
C                                       ***RESO N M****
       WRITE(8,145)IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IREMX
             WRITE (9,40) ARCID,IDOUT,REMAIN
             UBKNT=UBKNT+1
             GO TO 300
230          CONTINUE
C      **CINC**
             WRITE (9,30) ARCID,NBRSPC(N),NAME(5),VALUE
             DEN=(YTOUR(N)*NRATIO)-1
```

```
            NUM=DEN-1
            IF (NUM.LE.0) GO TO 280
            IEND=NUM
            IF (IEND.GE.NYRS) IEND=NYRS-1
            IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
            IDNEXT=IDOUT+10000
            REMAIN=1.
            VALUE1=REMAIN*(1./(YTOUR(N)*NRATIO))
            VALUE2=VALUE*SURVHI(1)*(1.-NUM/DEN)
C           ****RESO N M****                  ***RES1 N M****
      IVAL1X = VALUE1*1000.
      WRITE(8,145) IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL1X
      IVAL2X = VALUE2*1000.
      WRITE(8,145) IDNEXT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL2X
            WRITE (9,120) ARCID,RES,IDOUT,VALUE1,RES,IDNEXT,VALUE2
               DO 260 J=1,IEND
               FRACT=NUM/DEN
               VALUE=VALUE*SURVHI(J)*FRACT
               IF (J.EQ.(NYRS-1)) GO TO 290
               IDOUT=((J+1)*10000)+NBRSPC(N)*100+NBRSPC(M)
               VALUE1=VALUE*SURVHI(J+1)
               VALUE2=VALUE*SURVHI(J+1)*(1.-(NUM-1)/(DEN-1))
               ID=J*100+NBRSPC(N)
C               ***CINC**                ***RES ***
      IVAL2X = VALUE2*1000.
      WRITE(8,145) IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL2X
               WRITE (9,30) ARCID,ID,NAME(5),VALUE,RES,IDOUT,VALUE2
               GO TO 250
C             *********CINC********
240            WRITE (9,50) ARCID,J,NBRSPC(N),NAME(5),VALUE
               GO TO 300
250            DEN=DEN-1
               NUM=NUM-1
260            CONTINUE
            GO TO 300
270         MRATIO=YUTIL(M,N)/10
            NRATIO=YUTIL(M,N)-MRATIO*10
            LASTGO(N,M)=(NRATIO*YTOUR(N))-1
            IF (LASTGO(N,M).GE.NYRS) LASTGO(N,M)=NYRS
            REMAIN=1.
            VALUE=REMAIN*(1.-(1./(YTOUR(N)*NRATIO)))
            GO TO 220
280         VALUE1=VALUE*SURVHI(1)
            IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
            IDNEXT=IDOUT+10000
C           ***RESO N M ***            ***RES1 N M*****
      IVAL1X = VALUE1*1000.
      WRITE(8,145) IDOUT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL1X
      IVAL2X = VALUE2*1000.
      WRITE(8,145) IDNEXT,NRATIO,MRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL2X
            WRITE (9,120) ARCID,RES,IDOUT,VALUE,RES,IDNEXT,VALUE1
            GO TO 300
290         UBKNT=UBKNT+1
            LASTGO(N,M)=-1
300         CONTINUE
310      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOCOLL
C     ****************************************************************
C     *                                                              *
C     *       SUBROUTINE: LOCOLL                                     *
C     *       CALLED BY:  LOCOL                                      *
C     *       CALLING  ARGUMENTS: NONE                               *
C     *       CALLED ROUTINES: NONE                                  *
C     *       OUTPUT FILES:                                          *
C     *                   9 - ODEQAUD01.                             *
C     *                   8 - ODSAPUD08.                             *
C     *       PURPOSE: DEFINES XOOOM,X-M-M,AND Y-M-M,ARCS            *
C     *                FOR LT SEGMENT.                               *
C     *       DATE: 15 APRIL 76                                      *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R. L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                8120 WOODMONT AVENUE                          *
C     *                BETHESDA,MARYLAND 20014                       *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
      INTEGER ARCID
      REAL LTCPT
C
C                         *****FORMAT STATEMENTS*****
C
10    FORMAT ('COLUMNS')
20    FORMAT (4X,'XOOO',J2,4X,'NO',J2,A4,2X,F12.3,3X,'NO',J2,A4,2X,
     1F12.3)
30    FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.3,3X,'N',J1,J2,A4,2X,
     1F12.3)
40    FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.3)
45    FORMAT (4X,'XOOO',J2,4X,'TOTAUTH ',2X,F12.9)
50    FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.3,3X,'N',J1,J2,A4,2X,
     1F12.3)
60    FORMAT (4X,'Y',J5,4X,'N',J1,J2,A4,2X,F12.3)
65    FORMAT('XO',2J2,1X,4X,9X,'OOO',J2,J4)
68    FORMAT('Y',J5,1X,4X,'OOO',J2,J4)
C
C
C     ****** GENERATE COLUMNS  CHAPTER*******
C      ********        *******         *******
      WRITE (9,10)
C         ****COMPUTING  INITIAL FLOWS
C         **ENTRIES FOR XOOOMM**
         DO 70 M=1,NSPEC
         REMAIN=1.-PRMT(1)
C            **GOTO**                        **NO M LINC**
         WRITE (9,20) NBRSPC(M),NBRSPC(M),NAME(3),PLUS1,NBRSPC(M),
     1   NAME(4),REMAIN
          IVALU1 = REMAIN*1000
         WRITE (8,65) NBRSPC(M),NBRSPC(M),NBRSPC(M),IVALU1
C        *** TOTAUTH ***
         WRITE (9,45) NBRSPC(M),PLUS1
70       CONTINUE
C
```

```
C******FOR INCUMBENT ARCS TO ---T,NYRS (LIEUTENANT'S) **X-MM**
C
      DO 90 J=1,NYRS
      K=J-1
        DO 80 M=1,NSPEC
        ARCID=(K*10000)+NBRSPC(M)*100+NBRSPC(M)
C       **NK  GOZO**                **NJ  GOZO**
        WRITE (9,30) ARCID,K,NBRSPC(M),NAME(3),MINUS1,J,NBRSPC(M),
     1  NAME(3),SURVLO(J)
C                                   **NK  LINC**
        WRITE (9,40) ARCID,K,NBRSPC(M),NAME(4),MINUS1
        IF (J.EQ.NYRS) GO TO 80
C
        REMAIN=SURVLO(J)*(1.-PRMT(J+1))
        LTCPT=SURVLO(J)*(PRMT(J+1))
C       **NJ  TREQ**                **NJ  LINC**
        WRITE (9,30) ARCID,J,NBRSPC(M),NAME(2),PLUS1,J,NBRSPC(M),
     1  NAME(4),REMAIN
C       **NJ  CINC**
        IF (J.GE.ICHG) WRITE (9,40) ARCID,J,NBRSPC(M),NAME(5),LTCPT
    IVALU1 = LTCPT*1000
        IF (J.GE.ICHG) WRITE (8,68) ARCID,NBRSPC(M),IVALU1
80      CONTINUE
90    CONTINUE
C
C******FOR INCUMBENT ARC TO-T,NYRS  -CPT'S  LT 8 YRS SERVICE -Y-M-M**
C
      DO 110 J=1,NYRS
      K=J-1
        DO 100 M=1,NSPEC
        ARCID=(K*10000)+NBRSPC(M)*100+NBRSPC(M)
C       **NK  GOZO**                **NJ  GOZO**
        WRITE (9,50) ARCID,K,NBRSPC(M),NAME(3),MINUS1,J,NBRSPC(M),
     1  NAME(3),SURVHI(J)
        REMAIN=SURVHI(J)*CPTREM(J)
C       **NJ  TREQ**
        IF (K.EQ.0) WRITE (9,60) ARCID,J,NBRSPC(M),NAME(2),PLUS1
        IF (K.EQ.0) GO TO 100
C                     **NK  CINC**               **NJ  CREQ**
        IF (ICHG .LT. NYRS .AND.
     *  J.EQ.NYRS) WRITE (9,50) ARCID,K,NBRSPC(M),NAME(5),
     1  MINUS1,J,NBRSPC(M),NAME(1),PLUS1
        IF (J.EQ.NYRS) GO TO 100
        REMAIN=SURVHI(J)*CPTREM(J)
C                     **NK  CINC**               **NJ  CINC**
        IF (J.GT.ICHG) WRITE (9,50) ARCID,K,NBRSPC(M),NAME(5),
     1  MINUS1,J,NBRSPC(M),NAME(5),REMAIN
C                     **NJ  CINC**
        IF (J.EQ.ICHG) WRITE (9,60) ARCID,J,NBRSPC(M),NAME(5),REMAIN
C                     **NJ  TREQ**               **NJ  CREQ*
        IF (J.GT.ICHG) WRITE (9,50) ARCID,J,NBRSPC(M),NAME(2),PLUS1,
     1  J,NBRSPC(M),NAME(1),PLUS1
C                     **NJ  TREQ**
        IF (J.LE.ICHG) WRITE (9,60) ARCID,J,NBRSPC(M),NAME(2),PLUS1
100     CONTINUE
110   CONTINUE
    RETURN
    END
```

```
      SUBROUTINE LOCOLS
C     ***************************************************************
C     ***************************************************************
C     *                                                             *
C     *       SUBROUTINE: LOCOLS                                    *
C     *       CALLED BY:  LOCOL                                     *
C     *       CALLING  ARGUMENTS: NONE                              *
C     *       CALLED ROUTINES: IBITS,SET,IPHASE,JPHASE,IPROB,OPT    *
C     *       OUTPUT FILES:                                         *
C     *                    4 - ODSAPUD04.                           *
C     *                    8 - ODSAPUD08.                           *
C     *                    9 - ODEQAUD01.                           *
C     *       PURPOSE: DEFINES XOOOM,WOM-M,WOM-M,AND X-M-M.         *
C     *                PRODUCES OPTIONAL OUTPUT FOR INDICATING      *
C     *                WHEN LAST OF T-O POPULATION IS REASSIGNED    *
C     *                FROM SPECIALTY M. ADDITIONALLY PRODUCES      *
C     *                UTILIZATION RATIO,TOUR LENGTH,AND FUNCTIONAL *
C     *                RELATIONSHIP DATA FOR INPUT TO DATA BASES.   *
C     *       DATE: 15 APRIL 76                                     *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                8120 WOODMONT AVENUE                         *
C     *                BETHESDA,MARYLAND 20014                      *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1
      INCLUDE PROC2
      INTEGER ARCID
      DIMENSION LSAVE(50)
C
C                   *****FORMAT STATEMENTS*****
C
10    FORMAT ('COLUMNS')
20    FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J1,J2,A4,2X,
     1F12.9)
30    FORMAT (4X,'W',J5,4X,'WO',J2,A4,2X,F12.9,3X,A1,J3,A4,2X,F12.9)
40    FORMAT (4X,'W',J5,4X,'N',J3,A4,2X,F12.9,3X,A3,J5,2X,F12.9)
41    FORMAT (4X,'W',J5,4X,'W',J3,A4,2X,F12.9,3X,A3,J5,2X,F12.9)
42    FORMAT (4X,'W',J5,4X,'W',J3,A4,2X,F12.9,3X,'OBJECTIV',2X,F12.9)
45    FORMAT (4X,'W',J5,4X,'TOTAUTH ',2X,F12.9)
50    FORMAT (4X,'W',J5,4X,'R',J5,4X,F12.9,3X,A1,J3,A4,2X,F12.9)
46    FORMAT (4X,'W',J5,4X,'UR',2J2,4X,F12.9)
65    FORMAT (25X,'LAST EXIT YEAR POINTS FOR GRADE',I3,/,'O SPECIALTY',
     * '    NOTE: POSITIVE VALUES INDICATE LAST YEAR WHEN T-O POPULAT',
     * 'ION DEPARTS A SPECIALTY ')
70    FORMAT(I8,5X,50I2)
80    FORMAT (4X,'W',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J5,4X,F12.9)
90    FORMAT (4X,'XOOO',J2,4X,'WO',J2,A4,2X,F12.9)
100   FORMAT (4X,'W',J5,4X,A1,J5,4X,F12.9,3X,A1,J5,4X,F12.9)
110   FORMAT (1H1)
121   FORMAT('X',J5,1X,4I1,9X,J5,J4)
122   FORMAT('Y',J5,1X,4I1,9X,J5,J4)
124   FORMAT('W',J5,1X,4I1)
C     ****** GENERATE COLUMNS  CHAPTER*******
C         ********          *****          *******
```

```
      WRITE (9,10)
C          ****COMPUTING COEFF FOR INITIAL FLOWS
C         **ENTRIES FOR XOOOMM**
        DO 140 M=1,NSPEC
        CALL IPHASE(M,$140)          @ SEGMENT
C              **GOZO**
        WRITE (9,90) NBRSPC(M),NBRSPC(M),NAME(3),PLUS1
        IF(IPROB(M).GT.0.AND.ISEG.LT.2)
C                    *****UBSG******
     *  WRITE (9,90) NBRSPC(M),NBRSPC(M),NAME(6),PLUS1
140     CONTINUE
C     **T-ZERO-PRIME VARIABLES-WOMMNN**
C
        DO 320 N=1,NSPEC
        CALL IPHASE(N,$320)
        UBKNT=0
          DO 310 M=1,NSPEC
        CALL IPHASE(M,$310)
        CALL JPHASE(M,N,$310)
C       **IF SHOULD NOT EXIST,COUNT TOWARDS UB ON N&SET LASTGO(N,M)=-1
          IF (UTIL(M,N).EQ.99) GO TO 290
C        ***FOR FIELD GRADES M TO N ARCS DO NOT EXIST IN T-ZERO PRIME
          IF (M.EQ.N) GO TO 290
          IF ((UTIL(M,N).EQ.88).AND.(UTIL(N,M).GE.88)) GO TO 290
          IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
          ARCID=(NBRSPC(M)*100)+NBRSPC(N)
C     **GOZO**                **GOZO**
          WRITE (9,30) ARCID,NBRSPC(M),NAME(3),MINUS1,NN,NBRSPC(N),
     1    NAME(3),PLUS1
          IF (JGRADE.EQ.6) GO TO 150
C                ***TREQ***             ****RESO N M ***
          WRITE (9,40) ARCID,NBRSPC(N),NAME(2),PLUS1,RES,IDOUT,PRMT(1)
      IPRMT = PRMT(1)*1000
         IR = YUTIL(M,N)
         IF(IR.EQ.88) IR = YUTIL(N,M)
         IRATIO = IR/10
         JRATIO = IR - IRATIO*10
      WRITE(8,122)IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IPRMT
          GO TO 160
C                ****TREQ****
150       WRITE (9,40) ARCID,NBRSPC(N),NAME(2),PLUS1
C            **** TOTAUTH ****
160   IF(ISEG.NE.1) WRITE (9,45) ARCID,PLUS1
          IF (ISEG.LT.2.AND.IPROB(N).GT.0)
C              *** UBSEG ***                ***OBJECTIV***
     *    WRITE (9,42) ARCID,NBRSPC(N),NAME(6),PLUS1,PLUS1
C
C       **CHECK IF COMBAT ARMS SPECIALTY(88) IS INVOLVED
          IF (UTIL(M,N).NE.88) GO TO 270
          IF (UTIL(N,M).GE.88) GO TO 290
          NRATIO=UTIL(N,M)/10
          MRATIO=UTIL(N,M)-NRATIO*10
          LASTGO(N,M)=NRATIO*TOUR(N)-1
          IF (LASTGO(N,M).GE.NYRS) LASTGO(N,M)=NYRS
          REMAIN=1.-PRMT(1)
          VALUE=REMAIN*(1.-(1./(TOUR(N)*NRATIO)))
170   CONTINUE          @ 8 MARCH 1976
```

```
C
      I1 = MIN (M,N)
      I1 = NBRSPC(I1)
      I2 = MAX (M,N)
      I2 = NBRSPC(I2)
C
      RATE= MRATIO
      IF (N.GT.M) RATE = - RATE
C       *** UR-----CONSTRAINT ***
      WRITE(9,46)ARCID,I1,I2,RATE
C      **VALUE = 0 IF TOUR(N)*NRATIO =1, ALL LEAVE AT T-ZERO
            IF (VALUE.NE.0.) GO TO 180
      IREMX = REMAIN*1000
      WRITE(8,121)IDOUT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IREMX
            INDEX=(N-1)*NSPEC+M
C                                          ***RD N M****
            IF (IBITS(RTYPE,INDEX).NE.2) WRITE (9,50) ARCID,IDOUT,REMAIN
            UBKNT=UBKNT+1
            GO TO 300
180         IF (JGRADE.EQ.6) GO TO 190
            VALUE1=VALUE*SURVLO(1)*PRMT(2)
            IDOUT=IDOUT+10000
C                **NO LINC***          ***RES1 N M***
            WRITE (9,40) ARCID,NBRSPC(N),NAME(4),VALUE,RES,IDOUT,VALUE1
      IPRMT = VALUE1*1000
         IR = YUTIL(M,N)
         IF(IR.EQ.88) IR = YUTIL(N,M)
         IRATIO = IR/10
         JRATIO = IR - IRATIO*10
      WRITE(8,122)IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IPRMT
            GO TO 200
C                          **LINC**
190         WRITE (9,40) ARCID,NBRSPC(N),NAME(4),VALUE
200         DEN=(TOUR(N)*NRATIO)-1
            NUM=DEN-1
            IF (NUM.LE.0) GO TO 280
            IEND=NUM
            IF (IEND.GE.NYRS) IEND=NYRS-1
            IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
            IDNEXT=IDOUT+10000
            REMAIN=1.-PRMT(1)
            VALUE1=REMAIN*(1./(TOUR(N)*NRATIO))
            VALUE2=VALUE*SURVLO(1)*(1.-PRMT(2))*(1.-NUM/DEN)
            INDEX=(N-1)*NSPEC+M
      IVAL1X = VALUE1*1000.
      WRITE(8,121) IDOUT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL1X
      IVAL2X = VALUE2*1000.
      WRITE(4,124) ARCID,MRATIO,NRATIO,TOUR(N),TOUR(M)
      WRITE(8,121) IDNEXT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL2X
            IF (IBITS(RTYPE,INDEX).EQ.2) GO TO 210
C           ****RD N M****                  ***R1 N M****
            WRITE (9,100) ARCID,RR,IDOUT,VALUE1,RR,IDNEXT,VALUE2
210            DO 260 J=1,IEND
               FRACT=NUM/DEN
               VALUE=VALUE*SURVLO(J)*(1.-PRMT(J+1))*FRACT
               IF (J.EQ.(NYRS-1)) GO TO 220
               IDOUT=((J+1)*10000)+NBRSPC(N)*100+NBRSPC(M)
```

```
          VALUE1=VALUE*SURVLO(J+1)*PRMT(J+1)
          VALUE2=VALUE*SURVLO(J+1)*(1.-PRMT(J+1))*(1.-(NUM-1)/
     1    (DEN-1))
          INDEX=((J+1)*NSPEC**2)+(N-1)*NSPEC+M
          IF (JGRADE.EQ.8) GO TO 230
          ID=J*100+NBRSPC(N)
C         ***LINC**                   ***RES***
          WRITE (9,40) ARCID,ID,NAME(4),VALUE,RES,IDOUT,VALUE1
      IPRMT = VALUE1*1000
        IR = YUTIL(M,N)
        IF(IR.EQ.88) IR = YUTIL(N,M)
        IRATIO = IR/10
        JRATIO = IR - IRATIO*10
      WRITE(8,122)IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IPRMT
C                                       **R  N M****
          IF (IBITS(RTYPE,INDEX).NE.2) WRITE (9,50) ARCID,IDOUT,
     1    VALUE2
      IVAL2X = VALUE2*1000.
           WRITE (8,121)
     *   IDOUT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL2X
          GO TO 250
C        *********LINC*******
220       WRITE (9,80) ARCID,J,NBRSPC(N),NAME(4),VALUE
          GO TO 300
230     IVAL2X = VALUE2*1000.
        WRITE(8,121) IDOUT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL2X
          IF (IBITS(RTYPE,INDEX).EQ.2) GO TO 240
C         *********LINC*********       **R  N M***
          WRITE (9,80) ARCID,J,NBRSPC(N),NAME(4),VALUE,RR,IDOUT,
     1    VALUE2
          GO TO 250
C              **** LINC ****
240       WRITE (9,80) ARCID,J,NBRSPC(N),NAME(4),VALUE
250       DEN=DEN-1
          NUM=NUM-1
260       CONTINUE
        GO TO 300
270     MRATIO=UTIL(M,N)/10
        NRATIO=UTIL(M,N)-MRATIO*10
        LASTGO(N,M)=(NRATIO*TOUR(N))-1
        IF (LASTGO(N,M).GE.NYRS) LASTGO(N,M)=NYRS
        REMAIN=1.-PRMT(1)
        VALUE=REMAIN*(1.-(1./(TOUR(N)*NRATIO)))
        GO TO 170
280     VALUE1=VALUE*SURVLO(1)*(1.-PRMT(2))
        INDEX=(N-1)*NSPEC+M
        IDOUT=(NBRSPC(N)*100)+NBRSPC(M)
        IDNEXT=IDOUT+10000
      WRITE(4,124) ARCID,MRATIO,NRATIO,TOUR(N),TOUR(M)
      IVAL3X = VALUE*1000.
      WRITE(8,121) IDOUT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL3X
      IVAL1X = VALUE1*1000.
      WRITE(8,121) IDNEXT,NRATIO,MRATIO,TOUR(N),TOUR(M),ARCID,IVAL1X
        IF (IBITS(RTYPE,INDEX).EQ.2) GO TO 310
C       ***RO N M ***            ***RI N M*****
        WRITE (9,100) ARCID,RR,IDOUT,VALUE,RR,IDNEXT,VALUE1
        GO TO 310
```

```
290         UBKNT=UBKNT+1
            LASTGO(N,M)=-1
300         IF (UBKNT.NE.NSPEC) GO TO 310
            INDEX=(N-1)*NSPEC+N
            CALL SET (RTYPE,INDEX,3)
310         CONTINUE
320       CONTINUE
C       **********      **********      **********      *******
C       WRITE OUT LAST YEAR EXIT POINTS FOR EACH SPECIALTY
C
      IF(IOPT(IRD).LT.1) GO TO 331
        WRITE(6,65)JGRADE
          DO 330 M=1,NSPEC
          CALL IPHASE(M,$330)
            DO 335 N=1,NSPEC
            CALL IPHASE(N,$335)
            CALL JPHASE(M,N,$335)
            LSAVE(N)=  MAXO(LASTGO(M,N),0)
335         CONTINUE
          WRITE (6,70) NBRSPC(M),(LSAVE(N),N=1,NSPEC)
330       CONTINUE
C       **********      **********      **********      *******
C      **** FOR M=N ,  X OMON, WHERE M=N INCUMBENTS**
C
331       DO 360 J=1,NYRS
          K=J-1
            DO 350 M=1,NSPEC
            CALL IPHASE(M,$350)
C            CALL JPHASE(M,M,$350)       @ PER THOMAS 15DEC
            ARCID=K*(10000)+(NBRSPC(M)*100)+NBRSPC(M)
C                  **GOZO**                       **GOZO**
            WRITE (9,20) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,NBRSPC(M)
     1      ,NAME(3),SURVLO(J)
            IF (J.EQ.NYRS) GO TO 340
C          ***TREG**                       **LINC***
            WRITE (9,20) ARCID,J,NBRSPC(M),NAME(2),PLUS1,NN,K,NBRSPC(M),
     1      NAME(4),MINUS1
            GO TO 350
C          *******LINC*****
340         WRITE (9,20) ARCID,K,NBRSPC(M),NAME(4),MINUS1
350         CONTINUE
360       CONTINUE
C       ****************************************************************
      WRITE (6,110)
      RETURN
      END
```

```
      SUBROUTINE LODIAG
C     ***************************************************************
C     ***************************************************************
C     *                                                             *
C     *     SUBROUTINE: LODIAG                                      *
C     *     CALLED BY: MAIN (IMMEDIATE RETURN FOR O-3'S AND BELOW)  *
C     *     CALLING  ARGUMENTS: NONE                                *
C     *     CALLED ROUTINES: IBITS, RES2GR,IPHASE,JPHASE            *
C     *     OUTPUT FILES:                                           *
C     *               9 - ODEQAUDOI.                                *
C     *     PURPOSE: DEFINES (X=M-N) AND THEIR COEFFICIENTS IN      *
C     *              APPROPRIATE CONSTRAINTS FOR COL/LTC/MAJ        *
C     *              SEGMENTS.                                      *
C     *     DATE: 15 APRIL 76                                       *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *              8120 WOODMONT AVENUE                           *
C     *              BETHESDA,MARYLAND 20014                        *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1
      INTEGER ARCID
C
C                         *****FORMAT STATEMENTS*****
C
10    FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J1,J2,A4,2X,
     1F12.9)
20    FORMAT (4X,'X',J5,4X,'R',J5,4X,F12.9,3X,A1,J1,J2,A4,2X,F12.9)
30    FORMAT (4X,'XN',J2,8X,'N',J1,J2,A4,2X,F12.9,3X,'OBJECTIV',2X,
     1F12.9)
40    FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A3,J5,2X,F12.9)
C
C     ****************************************************************
      IF (JGRADE.LE.3) RETURN
      HOLD1=1
      HOLD2=1
C        ***COMPUTING COEFF FOR FLOWS LEAVING SPEC M**
C           **********          *********          *********
         DO 300 J=1,NYRS
         K=J-1
            DO 290 M=1,NSPEC
            CALL IPHASE(M,$290)
               DO 280 N=1,NSPEC
               CALL IPHASE(N,$280)
               CALL JPHASE(M,N,$280)
               IF (M.EQ.N) GO TO 280
               INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               NITOUR=0
               ARCID=K*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
C     TO CONSIDER THOSE LEAVING SPEC M THE FIRST TIME
               IF (K.GT.LASTGO(M,N)) GO TO 240
C      *****IF NOT  PREFERRED, THEN ARC NOT CREATED***
               IF (UTIL(N,M).EQ.99) GO TO 280
C       ***IS M A COMBAT ARM  (N,M)=88
               IF (UTIL(N,M).NE.88) GO TO 50
```

```
C          ***ARE TWO COMBAT ARMS INVOLVED***
                 IF (UTIL(M,N).GE.88) GO TO 280
                 MRATIO=UTIL(M,N)/10
                 NRATIO=UTIL(M,N)-(MRATIO*10)
                 GO TO 60
50               NRATIO=UTIL(N,M)/10
                 MRATIO=UTIL(N,M)-NRATIO*10
C      ENTER THE GOZINTA AND GOZOUTA VALUES
C                  **GOZO**                **GOZO**
60               WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,
     1           NBRSPC(N),NAME(3),SURVLO(J)
                 IF (IBITS(RTYPE,INDEX).NE.2) GO TO 70
C                                        ****TREQ****
                 IF (J.NE.NYRS) WRITE (9,40) ARCID,J,NBRSPC(N),NAME(2),
     1           PLUS1
                 GO TO 90
70               IF (J.EQ.NYRS) GO TO 80
C           ***  RQ M N  ***                      *** TREQ ***
                 WRITE (9,20) ARCID,ARCID,MINUS1,NN,J,NBRSPC(N),NAME(2),
     1           PLUS1
                 GO TO 90
C                  *** RQ M N ***
80               WRITE (9,20) ARCID,ARCID,MINUS1
90               IF ((JGRADE.NE.6).AND.(J.NE.NYRS)) CALL RES2GR
     1           (J,M,N,MRATIO,NRATIO)
                 LENGTH=TOUR(N)*NRATIO
                 KT=K
                 IF ((KT+LENGTH).GE.NYRS) GO TO 210
C         ***THERE WILL BE FLOW OUT OF SPEC N IN KT + LENGTH**
                 KPER=KT+LENGTH
C      ***HOW MANY LINC CONSTRAINTS IN TOUR LENGTH
                 HOLD1=1.0
100              IEND=LENGTH-1
                 IF (IEND.GT.0) GO TO 110
C          **ONE YEAR TOUR***
                 NITOUR=NITOUR+1
                 IF (NITOUR.GT.1) GO TO 200
                 KPER2=K
                 GO TO 140
110                 DO 120 NI=1,IEND,2
                    KPER1=K+NI
                    KPER2=KPER1+1
                    VALUE1=HOLD1*SURVLO(KPER1)*(1-PRMT(KPER1+1))
                    VALUE2=VALUE1*SURVLO(KPER2)*(1-PRMT(KPER2+1))
                    IF ((NI+1).GT.IEND) GO TO 130
C                   ****LINC**                ****LINC****
                    WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(4),VALUE1,NN,
     1              KPER2,NBRSPC(N),NAME(4),VALUE2
120                 CONTINUE
                 HOLD1=VALUE2
                 GO TO 140
C                **LINC***
130              WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(4),VALUE1
                 HOLD1=VALUE1
                 KPER2=KPER1
C         IS THERE TIME TO ROTATE BACK TO SPEC M
140              IF ((KPER2+1).GE.NYRS) GO TO 270
```

```
C       RESTRICTIVE FLOW IN PERIOD KPER, N TO M
              INDEX=(KPER*NSPEC**2)+(N-1)*NSPEC+M
              HOLD1=HOLD1*SURVLO(KPER)*(1-PRMT(KPER+1))
              IF (IBITS(RTYPE,INDEX).EQ.2) GO TO 150
              IDOUT=KPER*(10000)+NBRSPC(N)*100+NBRSPC(M)
C         ****R-N-M*****
              WRITE (9,20) ARCID,IDOUT,HOLD1
C     ROTATE TO M AND DETERMINE TIME IN SPEC M
150           LENGTH=MRATIO*TOUR(M)
              IF ((KPER+LENGTH-1).GE.NYRS) LENGTH=NYRS-KPER
C      ***FULL TOUR
C     ****HOW MANY LINC CONSTRAINTS FOR SPEC M
              IEND=LENGTH-1
              IF (IEND.GT.0) GO TO 160
C           ***ONE YEAR TOUR**
              KPER2=KPER2+1
              GO TO 190
160              DO 170 MI=1,IEND,2
                 KPER1=KPER+MI
                 KPER2=KPER1+1
                 VALUE1=HOLD1*SURVLO(KPER1)*(1-PRMT(KPER1+1))
                 VALUE2=VALUE1*SURVLO(KPER2)*(1-PRMT(KPER2+1))
                 IF ((MI+1).GT.IEND) GO TO 180
C                ***LINC**              **LINC**
                 WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(4),VALUE1,NN,
     1           KPER2,NBRSPC(M),NAME(4),VALUE2
                 HOLD1=VALUE2
170              CONTINUE
             GO TO 190
C                     ***LINC****
180           WRITE (9,10) ARCID,KPER1,NBRSPC(M),NAME(4),VALUE1
              HOLD1=VALUE1
              KPER2=KPER1+1
C       ***ARE THERE ANY PERIODS BEYOND KPER2
190           IF ((KPER2+1).GE.NYRS) GO TO 270
C     ****** ROTATE TO SPEC N *******
              K=KPER2+1
C          **ADD THE LENGTH OF M***
              KPER=KPER+LENGTH
              IF (KPER.GE.NYRS) GO TO 270
              IDOUT=(KPER)*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
              HOLD1=HOLD1*SURVLO(KPER)*(1-PRMT(KPER+1))
              VALUE1=HOLD1
C       ***** WRITE OUT RESTRICTIVE FLOW FOR PERIOD KPER
C                                              ****R--M-N*****
              INDEX=(KPER*NSPEC**2)+(M-1)*NSPEC+N
              IF (IBITS(RTYPE,INDEX).NE.2) WRITE (9,20) ARCID,IDOUT,
     1        VALUE1
              LENGTH=TOUR(N)*NRATIO
C            ***ADD THE LENGT OF N***
              KPER=KPER+LENGTH
C     **IS THERE TIME TO STAY IN N FOR 1 MORE YR**
              K=K+1
              IF ((K).LT.NYRS) GO TO 100
              GO TO 270
C       **SPEC N IS ONE YEAR LONG**
200           IDOUT=(KPER*10000)+(NBRSPC(N)*100)+NBRSPC(M)
```

```
          HOLD1=HOLD1*SURVLO(KPER)*(1.-PRMT(KPER+1))
          VALUE1=HOLD1
C                                         ***R--N-M****
          INDEX=(KPER*NSPEC**2)+(N-1)*NSPEC+M
          IF (IBITS(RTYPE,INDEX).NE.2) WRITE (9,20) ARCID,IDOUT,
     1    VALUE1
          KPER2=KPER
          GO TO 140
C      **TIME IN SPEC N EXCEEDS HORIZON-WILL NOT LEAVE N***
210       IF (K.EQ.(NYRS-1)) GO TO 280
          RLEFT=SURVLO(J)*(1.-PRMT(J+1))
C                        **LINC***
          WRITE (9,10) ARCID,J,NBRSPC(N),NAME(4),RLEFT
C       **HOW MANY MORE LINCS***
          IEND=NYRS-(K+2)
          IF (IEND.LE.0) GO TO 280
             DO 230 L=1,IEND,2
             KPER1=K+L+1
             KPER2=KPER1+1
             VALUE1=RLEFT*SURVLO(KPER1)*(1.-PRMT(KPER1+1))
             VALUE2=VALUE1*SURVLO(KPER2)*(1.-PRMT(KPER2+1))
             IF ((L+1).GT.IEND) GO TO 220
C              ***LINC**                ***LINC***
             WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(4),VALUE1,NN,
     1       KPER2,NBRSPC(N),NAME(4),VALUE2
             RLEFT = VALUE2
             GO TO 230
C           ***LINC***
220          WRITE (9,10) ARCID,KPER1,NBRSPC(N),NAME(4),VALUE1
230          CONTINUE
          GO TO 280
C       ***K .GT. LASTGO(M)  ***
C     ***IF THERE IS NO RESTRICTIVE FLOW, THEN ARC NOT CREATED***
240       IF ((IBITS(RTYPE,INDEX).EQ.3).OR.(IBITS(RTYPE,INDEX)
     1    .EQ.0)) GO TO 280
C               **GOZO**                     **GOZO**
          WRITE (9,10) ARCID,K,NBRSPC(M),NAME(3),MINUS1,NN,J,
     1    NBRSPC(N),NAME(3),SURVLO(J)
          IDOUT=K*(10000)+(NBRSPC(M)*100)+NBRSPC(N)
          INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
          IF (J.NE.NYRS) GO TO 250
C                               ******R--M-N*****
          IF (IBITS(RTYPE,INDEX).NE.2) WRITE (9,20) ARCID,IDOUT,
     1    MINUS1
          GO TO 280
250       IF (IBITS(RTYPE,INDEX).EQ.2) GO TO 260
C           **R--M-N**                ***TREQ***
          WRITE (9,20) ARCID,IDOUT,MINUS1,NN,J,NBRSPC(N),NAME(2),
     1    PLUS1
          GO TO 280
C                                         ***TREQ***
260       WRITE (9,10) ARCID,J,NBRSPC(N),NAME(2),PLUS1
          GO TO 280
270       K=KT
280       CONTINUE
290    CONTINUE
300   CONTINUE
```

```
C         ***********          *******          ********
C            *****WRITE CONSTRAINTS FOR XN--****
C
          DO 310 J=1,NSPEC
          CALL IPHASE(J,$310)
C          ******N MMGOZO*****          ****OBJECTIV****
          WRITE (9,30) NBRSPC(J),NYRS,NBRSPC(J),NAME(3),MINUS1,PLUS1
310       CONTINUE
C         ********          *******          *********
      END
```

```
C     ***********************************************************************
C     *                                                                     *
C     *         PROGRAM: MAIN                                               *
C     *         CALLED BY: NONE                                             *
C     *         CALLING  ARGUMENTS: NONE                                    *
C     *         CALLED ROUTINES:  INPUT,ROWCHP,RESLO,ROWOP                  *
C     *                           RESHI,LOCOL,LODIAG,HICOL                  *
C     *                           RHS,RANGE,BOUNDS,MASK,KEYARC              *
C     *         PURPOSE: THE ROUTINE IS THE EXECUTIVE FOR THE               *
C     *                  MATRIX GENERATOR.                                  *
C     *         DATE: 15 APRIL 76                                           *
C     *         AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN      *
C     *         COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                 *
C     *                  8120 WOODMONT AVENUE                               *
C     *                  BETHESDA,MARYLAND 20014                            *
C     *                                                                     *
C     ***********************************************************************
C
C
      CALL INPUT
      CALL ROWCHP        @GENERATES ROW CHAPTER
      CALL KEYARC        @
      CALL RESLO         @
      CALL ROWOP         @
      CALL RESHI         @
      CALL LOCOL         @GENERATES COLUMN CHAPTER
      CALL LODIAG        @
      CALL HICOL         @
      CALL RHS           @ GENERATES RIGHT HAND SIDE CHAPTER
      CALL RANGE         @ GENERATES RANGE CHAPTER
      CALL BOUNDS        @ GENERATES BOUNDS CHAPTER
      CALL MASK          @ GENERATES MASK   CHAPTER
      CALL SORTW         @SORTS DATA ON W-ARCS FOR DATABASE CREATION
      CALL SORTXY        @SORTS DATA ON X-ARCS FOR DATABASE CREATION
      STOP
      END
```

```
      SUBROUTINE MASK
C
C     *****************************************************************
C     *                                                               *
C     *     SUBROUTINE: MASK                                          *
C     *     CALLED BY: MAIN                                           *
C     *     CALLING ARGUMENTS: NONE                                   *
C     *     CALLED ROUTINES: NONE                                     *
C     *     OUTPUT FILES: ODEQAUDDI. - 9 -                            *
C     *     PURPOSE: WRITES OUT THE MASK DATA REQUIRED FOR            *
C     *              THE LOADLIST PROCEDURE IN FMPS.                  *
C     *              IT GENERATES THE STANDARD MASKS USED FOR         *
C     *              ODDBSUDI. AND ODSOLUDI. SEGMENT DEPENDENT MASKS  *
C     *              ARE IN GRADE SEGMENT RUNSTREAMS                  *
C     *     DATE: 15 APRIL 76                                         *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN    *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY               *
C     *              8120 WOODMONT AVENUE                             *
C     *              BETHESDA,MARYLAND 20014                          *
C     *                                                               *
C     *****************************************************************
C
      INCLUDE PROC1,LIST
1     FORMAT('ENDATA')
2     FORMAT('NAME',10X,'MASKDATA')
3     FORMAT(8X,'N***TREQ')
4     FORMAT('NAME',10X,'MASKDATH')
5     FORMAT('MASKS',/8X,'XN**')
6     FORMAT('MASKS',/8X,'AA**')
7     FORMAT('MASKS',/8X,'N***CREQ')
8     FORMAT('MASKS',/8X,'X000**',2X,'XN**')
      WRITE(9,1)
C     MASKS FOR FMPS LOADLIST(LISTR) FOR SOLUTION FILE
      WRITE(9,2)
      WRITE(9,7)
      WRITE(9,3)
      WRITE(9,1)
      WRITE(9,4)
      IF(JGRADE.NE.2)WRITE(9,5)
      IF(JGRADE.EQ.2)WRITE(9,8)
      WRITE(9,1)
      WRITE(9,4)
      WRITE(9,6)
      WRITE(9,1)
      ENDFILE 9
      ENDFILE 8
      ENDFILE 4
      RETURN
      END
```

```
*   *********************************************************************
*   *                                                                   *
*   *       SUBROUTINE: OPT                                             *
*   *       CALLED BY: LINKAGE,ROWOP                                    *
*   *       PURPOSE: USED TO CONTROL PRINTING OF OPTIONAL OUTPUT.       *
*   *                IF AN OPTION APPEARS ON THE XQT CARD,THE           *
*   *                OUTPUT WILL BE PRODUCED,OTHERWISE NOT.             *
*   *       DATE: 15 APRIL 76                                           *
*   *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN      *
*   *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                 *
*   *                8120 WOODMONT AVENUE                               *
*   *                BETHESDA,MARYLAND 20014                            *
*   *                                                                   *
*   *********************************************************************
          AXR$ .
WB        EQU       1.
$(1)      .         CALL OPT(VALUE)
OPT*      ER        OPT$                  . GET OPTION WORD
          LA,U      A1,*0,X11             .
          SA        A0,0,A1               . STORE IT IN VALUE
          J         1+WB,X11              . RETURN
          END
```

```
C       ****************************************************************
C       *                                                              *
C       *         PROCEDURE: PROC                                      *
C       *         CALLED BY: ALL MATRIX GENERATOR SUBROUTINES          *
C       *         PURPOSE: TO DEFINE COMMON VARIABLES AND ARRAYS.      *
C       *         DATE: 15 APRIL 76                                    *
C       *         AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C       *         COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY          *
C       *                  8120 WOODMONT AVENUE                        *
C       *                  BETHESDA,MARYLAND 20014                     *
C       *                                                              *
C       ****************************************************************
PROC1   PROC
C
        COMMON /BLOCK/ MINUS1,       NAME(6),       NBRSPC(50),
       1 NN,           PLUS1,        RES,           RR,
       2 ZERO,         MAXSPC,       MAXYRS,        IVALUE
C
        COMMON /INPTX/ AES(50),      CPTREM(9),     ICHG,
       1 JGRADE,       NBRAES,       NSPEC,         NYRS,
       2 OFLOHI,       OFLOLO,       PRMT(9),       TOUR(50),
       3 UFLOHI,       UFLOLO,       UTIL(50,50),   YTOUR(50),
       4 YUTIL(50,50),NBRPRO,        NPROB(50),      UPBND(50),
       5 SARRAY(50),   ISEG,         AUTHMX(6)
C
        COMMON /COMPU/ LASTGO(50,50),REQ1(500),     REQ2(500),
       1 ROWCNT,       SURVHI(9),    SURVLO(9),     IROWCT(7)
C
        COMMON /SETXA/  RESFLO(2084),      RTYPE(2084)
C
         REAL MINUS1
        INTEGER AES,ROWCNT,REQ1,REQ2,TOUR,YTOUR,UTIL,YUTIL
  END
PROC2   PROC
        DEFINE IOPT(NN)       = AND(IVALUE,2**(1RZ-NN))
  END
```

```
C     ***************************************************************
C     *                                                             *
C     *      SUBROUTINE: RANGE                                      *
C     *      CALLED BY: MAIN                                        *
C     *      CALLING  ARGUMENTS: NONE                               *
C     *      CALLED ROUTINES: IPHASE,JPHASE,VALID                   *
C     *      OUTPUT FILES:                                          *
C     *                 9 - ODEQAUD01.                              *
C     *      PURPOSE: THE PURPOSE OF THE SUBROUTINE IS TO           *
C     *               SPECIFY THE RANGES FOR THE TREQ CONSTRAINTS   *
C     *               IN THE CPT SEGMENT        . RANGES ARE ALSO   *
C     *               COMPUTED FOR THE UR----CONSTRAINTS.           *
C     *      DATE: 15 APRIL 76                                      *
C     *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *               8120 WOODMONT AVENUE                          *
C     *               BETHESDA,MARYLAND 20014                       *
C     *                                                             *
C     ***************************************************************
      SUBROUTINE RANGE
      INCLUDE PROC1
      INTEGER HILIM
      REAL LORANG
C
C                        *****FORMAT STATEMENTS*****
C
10    FORMAT ('RANGES')
20    FORMAT (4X,'REQBAND',3X,'N1',J2,A4,2X,F12.3)
25    FORMAT (4X,'REQBAND',3X,'UR',2J2,4X,F12.3)
30    FORMAT (4X,'REQBAND',3X,'N',I1,J2,A4,2X,F12.3)
C
C      *****GENERATE RANGE CHAPTER****
C           ****************************************
      IF (JGRADE.LT.3) RETURN
      WRITE (9,10)
      IEND=NYRS+1
      IF(NSPEC.LT.16) GO TO 998  @ RANGE DOES NOT APPLY TO SMALL PROBLEMS
      IF (JGRADE.NE.3) GO TO 60
C
C    FOR T0 - T(NYRS-1) THE REQ1 VALUES FOR GRADE OF MAJOR (UNFILLED
C    REQUIREMENTS) BECOME THE LOWER BOUNDS FOR THE TREQ CONSTRAINTS.
C
         DO 50 J=1,NYRS
         K=J-1
          IF(K.EQ.0)PRODR8=MIN(SURVHI(J),SURVLO(J))
          IF(K.GT.0)PRODR8=PRODR8*MIN(SURVHI(K),SURVLO(K))
          DECR8=1.D - PRODR8
            DO 40 M=1,NSPEC
            INDEX=(M-1)*IEND+J
C         FOR AES, ESTABLISH AN HIGHER LOWER BOUND THAN FOR BES
C            TO REFLECT U/F MAJOR REQUIREMENTS PLUS MOST OF CPT
C            REQUIREMENTS
          IF(VALID(M).GT.0)LORANG=FLOAT(REQ2(INDEX)*DECR8)
           IF(VALID(M).GT.0) WRITE (9,30) K,NBRSPC(M),NAME(2),LORANG
40          CONTINUE
```

```
50        CONTINUE
C
60     IF (NSPEC .NE. 50) GO TO 998
          DO 70 J=1,NSPEC
          CALL IPHASE(J,$70)
          CALL JPHASE(J,J,$70)
          INDEX=(J-1)*IEND+2
          LOTOTL=REQ2(INDEX)-REQ1(INDEX)
          LOLIM=(FLOAT(LOTOTL)/(1.+OFLOLO))*(1.-UFLOLO)+.5
          HILIM=(FLOAT(REQ1(INDEX))/(1.+OFLOHI))*(1.-UFLOHI)+.5
          LOLIM=FLOAT(LOLIM)+FLOAT(HILIM)
          LORANG=FLOAT(REQ2(INDEX))-LOLIM
          WRITE (9,20) NBRSPC(J),NAME(2),LORANG
70        CONTINUE
998    IF(JGRADE.LT.3) RETURN
C      COMPUTE RANGES FOR KEY ARC RELATIONSHIP CONSTRAINTS
       NSPECX = NSPEC - 1
       DO 100  M = 1,NSPECX
       CALL IPHASE(M,$100)
       MM = M + 1
       DO 101  N = MM,NSPEC
       CALL IPHASE(N,$101)
       CALL JPHASE(M,N,$101)
          IF(JGRADE .GT. 3) GO TO 150
       IF(YUTIL(M,N) .LT.88) GO TO 200
       IF(YUTIL(M,N).EQ.88.AND.YUTIL(N,M).LT.88) GO TO 200
       GO TO 101
150      IF(UTIL(M,N) .LT.88) GO TO 200
       IF(UTIL(M,N).EQ.88.AND.UTIL(N,M).LT.88) GO TO 200
       GO TO 101
200    INDEX = (M-1)*IEND +1
       JINDEX = (N-1)*IEND +1
       VALUE1 = FLOAT(REQ2(INDEX))
       VALUE2 = FLOAT(REQ2(JINDEX))
C
C         BI EQUALS 5 PERCENT OF AVERAGE OF REQUIREMENTS
C
       BI = (VALUE1 + VALUE2) * 0.025
C
C         IF BI GE SMALLEST REQUIREMENT, SET BI=HALF OF SMALLEST REQ
       IF (BI .LT. MIN (VALUE1,VALUE2))GO TO 201
       BI = .5 *  MIN (VALUE1,VALUE2)
201    CONTINUE
       BI = BI*2.
       WRITE(9,23)NBRSPC(M),NBRSPC(N),BI
101    CONTINUE
100    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE RESHI
C     ..............................................................
C     ..............................................................
C     .                                                            .
C     .       SUBROUTINE: RESHI                                    .
C     .       CALLED BY: MAIN (IMMEDIATE RETURN FOR COL SEGMENT)   .
C     .       CALLING  ARGUMENTS: NONE                             .
C     .       CALLED ROUTINES: IBITS,SET,VALID,IPHASE,JPHASE,OPT   .
C     .       OUTPUT FILES:                                        .
C     .                     6 - STANDARD OUTPUT ON PRINTER         .
C     .                     9 - ODEQAUDOI.                         .
C     .       PURPOSE: THE SUBROUTINE DEFINES THE FLOW CONTROL     .
C     .                CONSTRAINTS FOR Y-ARCS FOR THE LT - LTC SEGMENTS.
C     .                FOR ALL APPROPRIATE YEARS. COMPUTES CODES FOR .
C     .                RESFLO ARRAY DEFINING THE EXISTANCE OF FLOW .
C     .                IN AN ARC,WHICH SPECIALTY IS PRIMARY AND IF .
C     .                A Y-ARC CONTAINS A FRACTION OF THE          .
C     .                FLOW ON AN EARLIER ARC IN THE NETWORK.      .
C     .       DATE: 15 APRIL 76                                    .
C     .       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN .
C     .       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY          .
C     .                8120 WOODMONT AVENUE                        .
C     .                BETHESDA,MARYLAND 20014                     .
C     .                                                            .
C     .                                                            .
C     ..............................................................
      INCLUDE PROC1
      INCLUDE PROC2
      INTEGER HIGRD
      INTEGER XLOS(50)
C
C                            .....FORMAT STATEMENTS.....
C
10    FORMAT (1X,'E',2X,'RES',J5)
20    FORMAT (1X,'L',2X,'RES',J5)
40    FORMAT (///,'1RESTRICTED FLOW CODES FOR GRADE',I3,2X,'YEAR T',I3,
     * /,' CODES: 0-NO FLOW  1-FLOW FROM PRIMARY',
     * ' 2-FLOW FROM ALTERNATE  3-FLOW FROM BOTH  4-LT SEGMENT-DESIGNATIONS +
50    FORMAT (1H0,'FOR SPECIALTY NO.',I3)
60    FORMAT (1H,50I2)
C
C     .....RESTRICTIVE FLOW CONSTRAINTS  ........
C        ...........               ...........
C      ....RESFLO COMES IN WITH '1' OR '2' OR '3' WHERE
C      ...THE LOWER GRADE WOULD BE PROMOTED - 1 -FROM PRIMARY
C      .........................'2'-PROMOTED FROM ALTERNATE.....
C     ..............'3' - PROMOTED FROM PRIMARY & ALTERNATE...
C     .............'0' - NO FLOW IN THE ARC....................
C        ...:. '4' - DESIGNATIONS AND FLOW FROM PREVIOUS ARC
C                  IN LT SEGMENT.........................
C
      IF (JGRADE.EQ.6) RETURN        @ NOT APPLICABLE TO COLONELS SEGMENT
         IF(JGRADE.GT.3)GO TO 130    @ BRANCH FOR LTC AND MAJ
C
C..... ASSIGNING PROMOTION FROM PRIMARY CODE TO BES FOR CPTS
```

```
C***** ASSIGNING PROMOTION FROM PRIMARY CODE TO BES FOR LTS AS FOLLOWS:
C        CODE 1 - BES TO ANY OTHER PREFRRRED SPECIALTY
C        CODE 1 - AES TO NON-COMBAT BES
C        CODE 2 - AES TO COMBAT BES
C         **AES TO AES NOT ALLOWED**
C
      IBGN=1
      IEND = MIN (ICHG,NYRS)
      IF (JGRADE.EQ.3) GO TO 70
      IF (ICHG.GE.NYRS) RETURN
C         ** +1 USED BECAUSE IN YEAR ICHG FOR LTS, FLOWS
C              ARE REQUIREMENTS DRIVEN - NO CONSTRAINTS**
      IBGN=ICHG+1
      IEND=NYRS
70       DO 120 J=IBGN,IEND
         K=J-1
           DO 110 M=1,NSPEC
           IF (VALID(M).GT.O.AND.JGRADE.GT.2) GO TO 110
              DO 100 N=1,NSPEC
              IF (M.EQ.N) GO TO 100
C*****  IF N IS COMBAT ARM AND NO PREFERENCE FOR M (EXIT DO LOOP)
              IF (YUTIL(M,N).GE.88.AND.YUTIL(N,M).GE.88) GO TO 100
C          IF M IS A COMBAT ARM AND N IS A PREFERENCE
              IF (YUTIL(N,M).EQ.88.AND.YUTIL(M,N).LT.88) GO TO 90
C           IF M IS A COMBAT ARM AND NO PREFERENCE FOR N
              IF (YUTIL(N,M).EQ.88.AND.YUTIL(M,N).GE.88) GO TO 100
C          **FOR LTS ONLY - AES TO CNBT ARM ALLOWED**
      IF (YUTIL(M,N).EQ.88.AND.YUTIL(N,M).LT.88 .AND. JGRADE .LT.3)
     * GO TO 95
C          IF COMBAT ARM IS ALTERNATE EXIT DO LOOP
              IF (YUTIL(M,N).EQ.88.AND.YUTIL(N,M).LT.88) GO TO 100
C          **IF N IS A COMBAT ARM AND NO PREFERENCE FOR M**
              IF (YUTIL(M,N).EQ.88.AND.YUTIL(N,M).GE.88) GO TO 100
C         **NEITHER M NOR N CMBT ARM, N IS PREF OF M**
              IF (YUTIL(M,N).LT.88) GO TO 90
              GO TO 100
90            INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
C          **FOR LTS ONLY - TWO AES MAY NOT BE PAIRED**
      IF(JGRADE.LT.3 .AND. VALID(M).GT.O .AND.VALID(N).GT.0)
     * GO TO 100
              CALL SET (RESFLO,INDEX,1) @ SET TO PRIMARY
      GO TO 100
95       INDEX=(K*NSPEC**2)+(M-1)*NSPEC +N
         CALL SET(RESFLO,INDEX,2)      @ SET TO ALT FOR LTS ONLY
100           CONTINUE
110        CONTINUE
120      CONTINUE
      IBGN=1
C
C***** PLOTTING PATHS OF PROMOTIONS TO DETERMINE RESFLO CODES
C
130   CONTINUE
      IBGN=1
      IEND=NYRS
      IF (JGRADE.LT.3) IBGN=ICHG+1
         DO 290 J=IBGN,IEND
         K=J-1
```

```
            DO 280 M=1,NSPEC
            CALL IPHASE(M,$280)
            IF (VALID(M).GT.0 .AND. JGRADE .EQ. 3) GO TO 280
               DO 270 N=1,NSPEC
               CALL IPHASE(M,$270)
               CALL JPHASE(M,N,$270)
               IF (M.EQ.N) GO TO 270
               MOVE=K
               SIGNAL='STOP'
               INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               IF (IBITS(RESFLO,INDEX).EQ.1) GO TO 210
C        ..FOR LTS ONLY - TREAT A 4 AS A 1 CODE..
               IF (IBITS(RESFLO,INDEX).EQ.4) GO TO 210
               IF (IBITS(RESFLO,INDEX).EQ.0) GO TO 270
C       ....RESFLO CODE IS '2' OR '3'  .....
               IF (IBITS(RESFLO,INDEX).EQ.3) SIGNAL='GO'
               NRATIO=YUTIL(N,M)/10
               MRATIO=YUTIL(N,M)-NRATIO*10
               MOVE=MOVE+NRATIO*YTOUR(N)
140            IF (MOVE.LE.(NYRS-1)) GO TO 150
               IF (SIGNAL.NE.'GO') GO TO 270
               MOVE=K
               GO TO 210
C.......        PRIMARY TO ALTERNATE
150            INDEX=(MOVE*NSPEC**2)+(N-1)*NSPEC+M
               IF (JGRADE.GT.2) GO TO 160
               IF (IBITS(RESFLO,INDEX).NE.0) CALL SET (RESFLO,INDEX,4)
               GO TO 170
160            CONTINUE
               IF ((IBITS(RESFLO,INDEX).NE.3).AND.(IBITS(RESFLO,INDEX)
     1         .NE.2)) CALL SET (RESFLO,INDEX,1)
               IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
170            MOVE=MOVE+MRATIO*YTOUR(M)
               IF (MOVE.LE.(NYRS-1)) GO TO 180
               IF (SIGNAL.NE.'GO') GO TO 270
               MOVE=K
               GO TO 210
C.......ALTERNATE TO PRIMARY
180            INDEX=(MOVE*NSPEC**2)+(M-1)*NSPEC+N
               IF (JGRADE.GT.2) GO TO 190
               IF (IBITS(RESFLO,INDEX).NE.0) CALL SET (RESFLO,INDEX,4)
               GO TO 200
190            CONTINUE
               IF ((IBITS(RESFLO,INDEX).NE.3).AND.(IBITS(RESFLO,INDEX)
     1         .NE.1)) CALL SET (RESFLO,INDEX,2)
               IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
200            MOVE=MOVE+NRATIO*YTOUR(N)
               GO TO 140
C       ...RESFLO CODE EQUALS PR.......
210            MRATIO=YUTIL(M,N)/10
               NRATIO=YUTIL(M,N)-MRATIO*10
               MOVE=MOVE+NRATIO*YTOUR(N)
220            IF (MOVE.GT.(NYRS-1)) GO TO 270
C.........ALTERNATE TO PRIMARY
               INDEX=(MOVE*NSPEC**2)+(N-1)*NSPEC+M
               IF (JGRADE.GT.2) GO TO 230
               IF (IBITS(RESFLO,INDEX).NE.0) CALL SET (RESFLO,INDEX,4)
```

```
                GO TO 240
230             CONTINUE
                IF ((IBITS(RESFLO,INDEX).NE.3).AND.(IBITS(RESFLO,INDEX)
     1          .NE.1)) CALL SET (RESFLO,INDEX,2)
                IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
240             MOVE=MOVE+MRATIO*YTOUR(M)
                IF (MOVE.GT.(NYRS-1)) GO TO 270
C*********PRIMARY TO ALTERNATE
                INDEX=(MOVE*NSPEC**2)+(M-1)*NSPEC+N
                IF (JGRADE.GT.2) GO TO 250
                IF (IBITS(RESFLO,INDEX).NE.0) CALL SET (RESFLO,INDEX,4)
                GO TO 260
250             CONTINUE
                IF ((IBITS(RESFLO,INDEX).NE.3).AND.(IBITS(RESFLO,INDEX)
     1          .NE.2)) CALL SET (RESFLO,INDEX,1)
                IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
260             MOVE=MOVE+YTOUR(N)*NRATIO
                GO TO 220
270             CONTINUE
280           CONTINUE
290        CONTINUE
        IF (JGRADE.NE.3) GO TO 370
C
C****** PLOTTING PATHS OF CPT/MAJ STARTING AT TO
C
           DO 360 M=1,NSPEC
              DO 350 N=1,NSPEC
C       ***K=0IS ASSUMMEDSTARTING TIME PERIOD***
              IF (M.EQ.N) GO TO 350
C          ***IS N/M PREFERRED  AT T-0,YES IF LT 99  ***
              IF (YUTIL(N,M).GE.99) GO TO 350
C          ***IS M A PROHIBITED SECONDARY***    YES IF EQUAL **
              IF (YUTIL(N,M).NE.88) GO TO 320
C       ***ARE M AND N BOTH COMBAT ARMS**
              IF (YUTIL(M,N).GE.88) GO TO 350
              MRATIO=YUTIL(M,N)/10
              NRATIO=YUTIL(M,N)-MRATIO*10
              LAST=MRATIO*YTOUR(M)
              IF (LAST.GT.NYRS) LAST=NYRS
C          **CONSIDER EACH SUB-GROUP OF T-ZERO POPULATION
                DO 310 L=1,LAST
                K=L-1
300             INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
                CALL SET (RESFLO,INDEX,1)
                K=K+YTOUR(N)*NRATIO
C          **IIF BEYOND HORIZON MOVE TO NEXT SUB-GROUP
                IF (K.GT.(NYRS-1)) GO TO 310
                INDEX=(K*NSPEC**2)+(N-1)*NSPEC+M
                CALL SET (RESFLO,INDEX,2)
                K=K+YTOUR(M)*MRATIO
                IF (K.GT.(NYRS-1)) GO TO 310
                GO TO 300
310             CONTINUE
              GO TO 350
320           NRATIO=YUTIL(N,M)/10
              MRATIO=YUTIL(N,M)-NRATIO*10
              LAST=MRATIO*YTOUR(M)
```

```
            IF (LAST.GT.NYRS) LAST=NYRS
              DO 340 L=1,LAST
              K=L-1
330           INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
              CALL SET (RESFLO,INDEX,2)
              K=K+YTOUR(N)*NRATIO
              IF (K.GT.(NYRS-1)) GO TO 340
              INDEX=(K*NSPEC**2)+(N-1)*NSPEC+M
              CALL SET (RESFLO,INDEX,1)
              K=K+YTOUR(M)*NRATIO
              IF (K.GT.(NYRS-1)) GO TO 340
              GO TO 330
340           CONTINUE
350         CONTINUE
360       CONTINUE
370     CONTINUE
        IBGN=1
        IF (JGRADE.EQ.2) IBGN=ICHG+1
          DO 420 J=IBGN,NYRS
          K=J-1
            DO 410 M=1,NSPEC
            CALL IPHASE(M,$410)
              DO 400 N=1,NSPEC
              CALL IPHASE(N,$400)
              CALL JPHASE(M,N,$400)
              INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
              IF (IBITS(RESFLO,INDEX).EQ.0) GO TO 400
              LOC=(K*10000)+(NBRSPC(M)*100)+NBRSPC(N)
              IF (JGRADE.NE.2) GO TO 380
              IF (IBITS(RESFLO,INDEX).LT.4) GO TO 400
C          ***WRITE .LT. CONSTRAINT FOR LTS***
              WRITE (9,20) LOC
              GO TO 390
380           CONTINUE
C
C          ***FOR CAPTS BEFORE J EXCEEDS ICHG AND FOR BES AND
C
C             N NOT A COMBAT ARM, WRITE A .LT. CONSTRAINT***
              IF (JGRADE.EQ.3.AND.J.LE.ICHG.AND.VALID(M).LT.1 .AND.
     1 YUTIL(M,N).NE. 88) WRITE(9,20) LOC
C          ***FOR CAPTS,BEFORE YEAR EXCEEDS ICHG,FOR BES,AND
C          *** N IS A COMBAT ARM, WRITE .EQ. CONSTRAINT***
              IF (JGRADE.EQ.3.AND.J.LE.ICHG.AND.VALID(M).LT.1 .AND.
     1 YUTIL(M,N).EQ. 88) WRITE(9,10) LOC
C          ***EQUALITY CONSTRAINT IF GRADE = 4 OR 5
C                              GRADE = 3 AND YEAR GT ICHG
C                              GRADE = 3 AND M IS AN AES
              IF (JGRADE.NE.3.OR.J.GT.ICHG.OR.VALID(M).GT.0) WRITE (9,
     1        10) LOC
390           CONTINUE
              IROWCT(7)  =IROWCT(7) + 1
400           CONTINUE
410         CONTINUE
420       CONTINUE
        IF (IOPT(IRD).LT.1) RETURN     @ OPTIONAL REPORT OF RESFLO CODES
        IBGN=1
        HIGRD=JGRADE+1
```

```
         DO 450 I=1,NYRS
         WRITE (6,40) MIGRD,I
            DO 440 J=1,NSPEC
            CALL IPHASE(J,$445)
            WRITE (6,50) NBRSPC(J)
            IEND=IBGN+(NSPEC-1)
               DO 430 M=IBGN,IEND
               XLOS(M-IBGN+1)=IBITS(RESFLO,M)
430            CONTINUE
            WRITE (6,60) (XLOS(M),M=1,NSPEC)
445         IBGN=IBGN+NSPEC
440         CONTINUE
450      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RESLO
C     ***************************************************************
C     ***************************************************************
C     *                                                             *
C     *      SUBROUTINE: RESLO                                      *
C     *      CALLED BY: MAIN (IMMEDIATE RETURN FOR O-3'S AND O-2'S) *
C     *      CALLING  ARGUMENTS: NONE                               *
C     *      CALLED ROUTINES: IBITS,SET,IPHASE,JPHASE               *
C     *      PURPOSE: COMPUTES CODES DEFINING WHETHER AN X-ARC      *
C     *               CONTAINS A FRACTION OF AN EARLIER ARC IN      *
C     *               THE NETWORK FOR ALL GRADES ABOVE CPT.         *
C     *      DATE: 15 APRIL 76                                      *
C     *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *               8120 WOODMONT AVENUE                          *
C     *               BETHESDA,MARYLAND 20014                       *
C     *                                                             *
C     ***************************************************************
      INCLUDE PROC1
C     *****RESTRICTIVE FLOW CONSTRAINTS  ********
C        ***********              ************
C        ********RESFLO CODES*****
C       *******  1  - PROMOTED FROM PRIMARY ******
C       *******  2  - PROMOTED FROM ALTERNATE ***
C       *******  3  - PROMOTIONS FROM BOTH SPECIALTIES ***
C
      IF (JGRADE.LE.3) RETURN     @ DOES NOT APPLY TO CAPTAINS OR BELOW
         DO 240 M=1,NSPEC
         CALL IPHASE(M,$240)
            DO 230 N=1,NSPEC
            CALL IPHASE(N,$230)
            CALL JPHASE(M,N,$230)
C     ***K=0IS ASSUMMEDSTARTING TIME PERIOD***
            IF (M.EQ.N) GO TO 230
C       ***IS N/M PREFERRED  AT T-0,YES IF LT 99  ***
            IF (UTIL(N,M).GE.99) GO TO 220
C       ***IS M A PROHIBITED SECONDARY***
            IF (UTIL(N,M).NE.88) GO TO 110
C     ***ARE M AND N BOTH COMBAT ARMS**
            IF (UTIL(M,N).GE.88) GO TO 220
            MRATIO=UTIL(M,N)/10
            NRATIO=UTIL(M,N)-MRATIO*10
            LAST=MRATIO*TOUR(M)
            IF (LAST.GT.NYRS) LAST=NYRS
C       **CONSIDER EACH SUB-GROUP OF T-ZERO POPULATION
               DO 100 L=1,LAST
               K=L-1
               KTEMP=K
C       ***KTEMP KEEPS TRACK OF ONE TOUR INCREMENTS***
10             INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               CALL SET (RTYPE,INDEX,1)
               IF (JGRADE.EQ.6) GO TO 20
C        ****PROMOTION FROM PRIMARY*****
               IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
               IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,1)
```

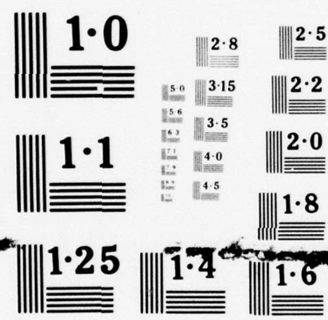
```
20             CONTINUE
               K=K+TOUR(N)*NRATIO
               IF (K.GT.(NYRS-1)) K=NYRS
C        **********,     *********      **********,      *******
C        KTEMP IS WHEN ONE TOUR LENGTH COMPLETED
               KTEMP=KTEMP+TOUR(N)
               IF (KTEMP.GT.(NYRS-1)) KTEMP=NYRS
C          ***IF THERE ARE MULTIPLE TOURS IN A SPEC, THEN AT THE END OF
C              EACH TOUR, PROMOTEES MOVE TO THEIR OTHER SPEC**
30             IF (KTEMP.EQ.K) GO TO 50
               INDEX=(KTEMP*NSPEC**2)+(N-1)*NSPEC+M
               IF (JGRADE.EQ.6) GO TO 40
C        *****PROMOTION FROM ATERNATE*****
               IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
               IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,2)
40             CONTINUE
               KTEMP=KTEMP+TOUR(N)
               IF (KTEMP.LE.(NYRS-1)) GO TO 30
C        **IF BEYOND HORIZON MOVE TO NEXT SUB-GROUP
50             IF (K.GT.(NYRS-1)) GO TO 100
               INDEX=(K*NSPEC**2)+(N-1)*NSPEC+M
               CALL SET (RTYPE,INDEX,1)
               IF (JGRADE.EQ.6) GO TO 60
C        *******PROMOTION FROM ALTERNATE*****
               IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
               IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,2)
60             CONTINUE
               K=K+TOUR(M)*MRATIO
               IF (K.GT.(NYRS-1)) K=NYRS
               KTEMP=KTEMP+TOUR(M)
               IF (KTEMP.GT.(NYRS-1)) KTEMP=NYRS
70             IF (KTEMP.EQ.K) GO TO 90
               INDEX=(KTEMP*NSPEC**2)+(M-1)*NSPEC+N
               IF (JGRADE.EQ.6) GO TO 80
C          *****PROMOTION FROM PRIMARY****
               IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
               IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,1)
80             CONTINUE
               KTEMP=KTEMP+TOUR(M)
               IF (KTEMP.LE.(NYRS-1)) GO TO 70
90             IF (K.GT.(NYRS-1)) GO TO 100
               GO TO 10
100            CONTINUE
           GO TO 230
110        NRATIO=UTIL(N,M)/10
           MRATIO=UTIL(N,M)-NRATIO*10
           LAST=MRATIO*TOUR(M)
           IF (LAST.GT.NYRS) LAST=NYRS
               DO 210 L=1,LAST
               K=L-1
               KTEMP=K
120            INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
               CALL SET (RTYPE,INDEX,1)
               IF (JGRADE.EQ.6) GO TO 130
C        *****PROMOTION FROM ALTERNATE*****
               IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
               IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,2)
```

AD-A040 832 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 5/9
OFFICER DUAL SPECIALTY ALLOCATION SYSTEM (ODSAS) DOCUMENTATION.(U)
MAY 77 J D THOMAS, G E ARMSTRONG

UNCLASSIFIED CAA-D-77-1 NL

5 OF 5
ADA
040832

END
DATE
FILMED
7-77


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

```
130       CONTINUE
          K=K+TOUR(N)*NRATIO
          IF (K.GT.(NYRS-1)) K=NYRS
          KTEMP=KTEMP+TOUR(N)
          IF (KTEMP.GT.(NYRS-1)) KTEMP=NYRS
140       IF (KTEMP.EQ.K) GO TO 160
          INDEX=(KTEMP*NSPEC**2)+(N-1)*NSPEC+M
          IF (JGRADE.EQ.6) GO TO 150
C     ***PROMOTION FROM PRIMARY*****
          IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
          IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,1)
150       CONTINUE
          KTEMP=KTEMP+TOUR(N)
          IF (KTEMP.LE.(NYRS-1)) GO TO 140
160       IF (K.GT.(NYRS-1)) GO TO 210
          INDEX=(K*NSPEC**2)+(N-1)*NSPEC+M
          CALL SET (RTYPE,INDEX,1)
          IF (JGRADE.EQ.6) GO TO 170
C      *****PROMOTION FROM PRIMARY*****
          IF (IBITS(RESFLO,INDEX).EQ.2) CALL SET (RESFLO,INDEX,3)
          IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,1)
170       CONTINUE
          K=K+TOUR(M)*MRATIO
          IF (K.GT.(NYRS-1)) K=NYRS
          KTEMP=KTEMP+TOUR(M)
          IF (KTEMP.GT.(NYRS-1)) KTEMP=NYRS
180       IF (KTEMP.EQ.K) GO TO 200
          INDEX=(KTEMP*NSPEC**2)+(M-1)*NSPEC+N
          IF (JGRADE.EQ.6) GO TO 190
C     *****PROMOTED FROM ALTERNATE*****
          IF (IBITS(RESFLO,INDEX).EQ.1) CALL SET (RESFLO,INDEX,3)
          IF (IBITS(RESFLO,INDEX).NE.3) CALL SET (RESFLO,INDEX,2)
190       CONTINUE
          KTEMP=KTEMP+TOUR(M)
          IF (KTEMP.LE.(NYRS-1)) GO TO 180
200       IF (K.GT.(NYRS-1)) GO TO 210
          GO TO 120
210       CONTINUE
        GO TO 230
220     INDEX=(M-1)*NSPEC+N
        CALL SET (RTYPE,INDEX,3)
230     CONTINUE
240   CONTINUE
C    **********     **********     **********     *******
     RETURN
     END
```

```
      SUBROUTINE RES2GR (J,M,N,MRATIO,NRATIO)
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *        SUBROUTINE: RES2GR                                    *
C     *        CALLED BY:  LODIAG                                    *
C     *        CALLING  ARGUMENTS:                                   *
C     *                             J - CURRENT YEAR                 *
C     *                             M - PRIMARY SPECIALTY            *
C     *                             N - ALTERNATE SPECIALTY          *
C     *                             MRATIO- UTIL. RATIO IN SPECIALTY M *
C     *                             NRATIO- UTIL. RATIO IN SPECIALTY N *
C     *        CALLED ROUTINES: NONE                                 *
C     *        OUTPUT FILES:                                         *
C     *                  8 - ODSAPDUO8.                              *
C     *                  9 - ODEQAUDO1.                              *
C     *        PURPOSE: DEFINES AN X-ARC'S FLOW UPON PROMOTION       *
C     *                 TO A HIGHER GRADE.                           *
C     *        DATE: 15 APRIL 76                                     *
C     *        AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *        COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *                 8120 WOODMONT AVENUE                         *
C     *                 BETHESDA,MARYLAND 20014                      *
C     *                                                              *
C     ****************************************************************
C
      INCLUDE PROC1
      INTEGER ARCID
      DIMENSION SAVAL(9)
C
C                          *****FORMAT STATEMENTS*****
C
121   FORMAT('Y',J5,1X,4I1,J5,I4)
C122   FORMAT('Y',J1,2J2,5X,J5,I4)
200   FORMAT (4X,'X',J5,4X,'N',J1,J2,A4,2X,F12.9,3X,A1,J1,J2,A4,2X,
     1F12.9)
400   FORMAT (4X,'X',J5,4X,A3,J5,2X,F12.9,3X,A3,J5,2X,F12.9)
C
         IR = YUTIL(M,N)
         IF(IR.EQ.88) IR = YUTIL(N,M)
         IRATIO = IR/10
         JRATIO = IR - IRATIO*10
      KPER=J
      ARCID=(J-1)*10000+NBRSPC(M)*100+NBRSPC(N)
      VALUE=1.0
600   IF ((NRATIO*TOUR(N)).NE.1) GO TO 1000
      IDOUT=(KPER*10000)+(NBRSPC(N)*100)+NBRSPC(M)
      VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
      VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
C              **RES(KPER)N M**
      WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVAL1X = VALUE1*1000
      WRITE(8,121) IDOUT,JRATIO,IRATIO,YTOUR(N),YTOUR(M),ARCID,IVAL1X
      KPER=KPER+1
      IF (KPER.GE.NYRS) RETURN
```

```
800   IF ((NRATIO*TOUR(M)).NE.1) GO TO 3200
      IDOUT=(KPER*10000)+NBRSPC(M)*100+NBRSPC(N)
      VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
      VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
C        **RES(KPER) M N**
      WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
      KPER=KPER+1
      IF (KPER.GE.NYRS) RETURN
      GO TO 600
1000  IF (NRATIO.NE.1) GO TO 1800
C        ***IEND = # OF CINC CONSTRAINTS***
      IEND=TOUR(N)-1
      IF ((KPER+IEND).GT.NYRS) IEND=NYRS-KPER
         DO 1400 INDEX=1,IEND
         VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
         SAVAL(INDEX)=VALUE1
         IENDS=KPER+IEND-INDEX
            DO 1200 K=KPER,IENDS
            SAVAL(INDEX)=SAVAL(INDEX)*SURVMI(K)
1200        CONTINUE
         VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
C        ***(KPER)-N-CINC***
         WRITE (9,200) ARCID,KPER,NBRSPC(N),NAME(5),VALUE1
      IVALIX = VALUE1*1000
C      WRITE(8,122)KPER,NBRSPC(N),NBRSPC(N),
C     * ARCID,IVALIX
         KPER=KPER+1
         IF (KPER.GE.NYRS) RETURN
1400     CONTINUE
      VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
         DO 1600 K=1,IEND
         VALUE1=VALUE1+SAVAL(K)
1600     CONTINUE
      VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
      IDOUT=(KPER*10000)+NBRSPC(N)*100+NBRSPC(M)
C        **RES(KPER) N-M**
      WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
      KPER=KPER+1
      IF (KPER.GE.NYRS) RETURN
C        ****NOW CHECK ON TOUR IN OTHER SPECIALTY****
      GO TO 800
C        *****NRATIO.NE. 1 AND NRATIO*TOUR(N).NE. 1 *****
1800     DO 3000 KK=1,NRATIO
         IEND=TOUR(N)
         IF (KK.EQ.1.OR.TOUR(N).EQ.1) IEND=TOUR(N)-1
         IF ((KPER+IEND).GT.NYRS) IEND=NYRS-KPER
         IF (IEND-1) 2000,2200,2200
2000     VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
         VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
         IDOUT=(KPER*10000)+NBRSPC(N)*100+NBRSPC(M)
C        *****RES (KPER)=N=M *****
         WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
```

```
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
        KPER=KPER+1
        IF (KPER.GE.NYRS) RETURN
        GO TO 3000
2200       DO 2600 INDEX=1,IEND
           VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
           SAVAL(INDEX)=VALUE1
           IENDS=KPER+IEND-INDEX
              DO 2400 K=KPER,IENDS
              SAVAL(INDEX)=SAVAL(INDEX)*SURVHI(K)
2400          CONTINUE
           VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
C        *****(KPER)-N-CINC*****
           WRITE (9,200) ARCID,KPER,NBRSPC(N),NAME(5),VALUE1
      IVALIX = VALUE1*1000
C      WRITE(8,122)KPER,NBRSPC(N),NBRSPC(N),ARCID,IVALIX
           KPER=KPER+1
           IF (KPER.GE.NYRS) RETURN
2600       CONTINUE
        VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
           DO 2800 K=1,IEND
           VALUE1=VALUE1+SAVAL(K)
2800       CONTINUE
        VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
        IDOUT=(KPER*10000)+NBRSPC(N)*100+NBRSPC(M)
C        ******RES (KPER)-N-M *****
        WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
        KPER=KPER+1
        IF (KPER.GE.NYRS) RETURN
3000    CONTINUE
C          ****NOW CHECK ON TOUR IN OTHER SPECIALTY****
      GO TO 800
3200  IF (MRATIO.NE.1) GO TO 4000
C       ******IEND=# OF CINC CONSTRAINTS***
      IEND=TOUR(M)-1
      IF ((KPER+IEND).GT.NYRS) IEND=NYRS-KPER
        DO 3600 INDEX=1,IEND
        VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
        SAVAL(INDEX)=VALUE1
        IENDS=KPER+IEND-INDEX
           DO 3400 K=KPER,IENDS
           SAVAL(INDEX)=SAVAL(INDEX)*SURVHI(K)
3400       CONTINUE
        VALUE=VALUE*SURVLO(KPER)*(1.-PRMT(KPER+1))
C      ***KPER-M-CINC****
        WRITE (9,200) ARCID,KPER,NBRSPC(M),NAME(5),VALUE1
      IVALIX = VALUE1*1000
C      WRITE(8,122)KPER,NBRSPC(M),NBRSPC(M),ARCID,IVALIX
        KPER=KPER+1
        IF (KPER.GE.NYRS) RETURN
3600    CONTINUE
      VALUE1=VALUE*SURVLO(KPER)*PRMT(KPER+1)
        DO 3800 K=1,IEND
        VALUE1=VALUE1+SAVAL(K)
3800    CONTINUE
```

```
      VALUE=VALUE*SURVLO(KPER)*(1.-PRHT(KPER+1))
      IDOUT=(KPER*10000)+NBRSPC(M)*100+NBRSPC(N)
C       ***RES KPER -M-N ****
      WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
      KPER=KPER+1
      IF (KPER.GE.NYRS) RETURN
C       ***NOW CHECK ONTOUR IN OTHER SPECIALTY****
      GO TO 600
4000     DO 5200 KK=1,MRATIO
         IEND=TOUR(M)
         IF (KK.EQ.1.OR.TOUR(M).EQ.1) IEND=TOUR(M)-1
         IF ((KPER+IEND).GT.NYRS) IEND=NYRS-KPER
         IF (IEND-1) 4200,4400,4400
4200     VALUE1=VALUE*SURVLO(KPER)*PRHT(KPER+1)
         VALUE=VALUE*SURVLO(KPER)*(1.-PRHT(KPER+1))
         IDOUT=(KPER*10000)+NBRSPC(M)*100+NBRSPC(N)
C      *****RES KPER -M-N ****
         WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
         KPER=KPER+1
         IF (KPER.GE.NYRS) RETURN
         GO TO 5200
4400        DO 4800 INDEX=1,IEND
            VALUE1=VALUE*SURVLO(KPER)*PRHT(KPER+1)
            SAVAL(INDEX)=VALUE1
            IENDS=KPER+IEND-INDEX
               DO 4600 K=KPER,IENDS
               SAVAL(INDEX)=SAVAL(INDEX)*SURVHI(K)
4600           CONTINUE
            VALUE=VALUE*SURVLO(KPER)*(1.-PRHT(KPER+1))
C           ***KPER M CINC****
            WRITE (9,200) ARCID,KPER,NBRSPC(M),NAME(5),VALUE1
      IVALIX = VALUE1*1000
C      WRITE(8,122)KPER,NBRSPC(M),NBRSPC(M),ARCID,IVALIX
            KPER=KPER+1
            IF (KPER.GE.NYRS) RETURN
4800        CONTINUE
         VALUE1=VALUE*SURVLO(KPER)*PRHT(KPER+1)
            DO 5000 K=1,IEND
            VALUE1=VALUE1+SAVAL(K)
5000        CONTINUE
         VALUE=VALUE*SURVLO(KPER)*(1.-PRHT(KPER+1))
         IDOUT=KPER*10000+NBRSPC(M)*100+NBRSPC(N)
C        ***RES KPER -M-N ****
         WRITE (9,400) ARCID,RES,IDOUT,VALUE1
      IVALIX = VALUE1*1000
      WRITE(8,121) IDOUT,IRATIO,JRATIO,YTOUR(N),YTOUR(M),ARCID,IVALIX
         KPER=KPER+1
         IF (KPER.GE.NYRS) RETURN
5200     CONTINUE
C        ***NOW CHECK FOR OTHER SPECIALTY ***
      GO TO 600
      END
```

```
      SUBROUTINE RHS
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *      SUBROUTINE: RHS                                         *
C     *      CALLED BY: MAIN                                         *
C     *      CALLING ARGUMENTS: NONE                                 *
C     *      CALLED ROUTINES: IPHASE,JPHASE                          *
C     *      OUTPUT FILES:                                           *
C     *                   9 - ODEQAUDOI.                             *
C     *      PURPOSE: THE SUBROUTINE GENERATES THE RIGHT HAND        *
C     *               SIDE CHAPTER OR THE B-VECTOR FOR THE MATRIX    *
C     *               GENERATOR'S CREQ, TREQ CONTRAINTS AND UBSG,    *
C     *               TOTAUTH,AND UR---- CONSTRAINTS.                *
C     *      DATE: 15 APRIL 76                                       *
C     *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *               8120 WOODMONT AVENUE                           *
C     *               BETHESDA,MARYLAND 20014                        *
C     *                                                              *
C     ****************************************************************
      INCLUDE PROC1
C
C                      *****FORMAT STATEMENTS*****
C
10    FORMAT ('RHS',11X,'B-VECTOR')
20    FORMAT (4X,'B-VECTOR',2X,'N',J1,J2,A4,2X,F12.3,3X,A1,J1,J2,A4,2X,
     1F12.3)
21    FORMAT (4X,'B-VECTOR',2X,'W',J1,J2,A4,2X,F12.3,3X,A1,J1,J2,A4,2X,
     1F12.3)
22    FORMAT (4X,'B-VECTOR',2X,'TOTAUTH ',2X,F12.3)
23    FORMAT (4X,'B-VECTOR',2X,'UR',2J2,4X,F12.3)
C
      IZERO  = 0
C
      WRITE (9,10)
      IEND=NYRS+1
C     RHS FOR CREQ AND TREQ CONSTRAINTS UP TO LAST YEAR
         DO 50 J=1,NYRS
         L=J-1
            DO 40 M=1,NSPEC
            CALL IPHASE(M,$40)      @ SEGMENT
            INDEX=(M-1)*IEND+J
            VALUE2=FLOAT(REQ2(INDEX))
            IF(ISEG.EQ.1) VALUE2 = VALUE2*SARRAY(M)
            IF (L.EQ.0) GO TO 30
            IF (JGRADE.EQ.8.OR.JGRADE.EQ.3) GO TO 30
            IF (JGRADE.EQ.2.AND.L.LE.ICHG) GO TO 30
            VALUE1=FLOAT(REQ1(INDEX))
            IF(ISEG.EQ.1) VALUE1 = VALUE1*SARRAY(M)
C           ******CREQ******             ******TREQ******
            WRITE (9,20) L,NBRSPC(M),NAME(1),VALUE1,NN,L,NBRSPC(M),
     1      NAME(2),VALUE2
            GO TO 40
C        ******TREQ******
```

```
30          IF (JGRADE.EQ.2.AND.L.EQ.0) GO TO 40
            WRITE (9,20) L,NBRSPC(M),NAME(2),VALUE2
40          CONTINUE
50       CONTINUE
      IF (JGRADE.EQ.6) GO TO 72           @ COL DO NOT REQUIRE CREQ VALUES
      IF (JGRADE.EQ.3) GO TO 95
      IF (JGRADE.EQ.2.AND.ICHG.GE.NYRS) GO TO 95
C     RHS FOR CREQ CONSTRAINTS IN LAST YEAR
         DO 60 M=1,NSPEC
         CALL IPHASE(M,$60)      @ SEGMENT
         INDEX=(M-1)*IEND+NYRS+1
         VALUE1=FLOAT(REQ1(INDEX))
            IF(ISEG.EQ.1) VALUE1 = VALUE1*SARRAY(M)
C           ****CREQ*****
         WRITE (9,20) NYRS,NBRSPC(M),NAME(1),VALUE1
60       CONTINUE
72    IF (ISEG .GT. 1.OR.JGRADE.LT.4) GO TO 80
C     RHS FOR CONTROL OF INPUT FOR SPECIFIED SPECIALTIES
      DO 79 M = 1,NBRPRO
      DO 73 K=1,NSPEC
      IF(NPROB(M).NE.NBRSPC(K)) GO TO 73
      J=K
      GO TO 74
73    CONTINUE
C        NO MATCH ON SPECIALTY NUMBER
       RETURN 0
C        *****UBSG*****
74    WRITE(9,21)IZERO,NPROB(M),NAME(6),UPBND(J)
79    CONTINUE
80    IF (ISEG.EQ.1) GO TO 96            @ 8 MARCH 76
C     RHS FOR TOTAL AUTHORIZED CONSTRAINTS
95    AUTH= AUTHMX(JGRADE)
      IF(JGRADE.LT.4) GO TO 91
      DO 90  M = 1,NBRPRO
      IF(ISEG.GT.1) AUTH = AUTH - UPBND(M)
90    CONTINUE
91    WRITE(9,22)AUTH
96    IF(JGRADE.LT.3) RETURN
C     RHS FOR KEY ARC RELATIONSHIP CONSTRAINTS
      NSPECX = NSPEC - 1
      DO 100  M = 1,NSPECX
      CALL IPHASE(M,$100)
      MM = M + 1
      DO 101  N = MM,NSPEC
      CALL IPHASE(N,$101)
      CALL JPHASE(M,N,$101)
         IF(JGRADE .GT. 3) GO TO 150
      IF(YUTIL(M,N) .LT.88) GO TO 200
      IF(YUTIL(M,N).EQ.88.AND.YUTIL(N,M).LT.88) GO TO 200
      GO TO 101
150     IF(UTIL(M,N) .LT.88) GO TO 200
      IF(UTIL(M,N).EQ.88.AND.UTIL(N,M).LT.88) GO TO 200
        GO TO 101
200   INDEX = (M-1)*IEND +1
      JINDEX = (N-1)*IEND +1
      VALUE1 = FLOAT(REQ2(INDEX))
      VALUE2 = FLOAT(REQ2(JINDEX))
```

```
C         BI EQUALS 5 PERCENT OF AVERAGE OF REQUIREMENTS
      BI = (VALUE1 + VALUE2) * 0.025
C
C         IF BI GE SMALLEST REQUIREMENT, SET BI=HALF OF SMALLEST REQ
      IF (BI .LT. MIN (VALUE1,VALUE2))GO TO 201
      BI = .5 *  MIN (VALUE1,VALUE2)
201   CONTINUE
      WRITE(9,23)NBRSPC(M),NBRSPC(N),BI
101   CONTINUE
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROWCHP
C     ********************************************************************
C     ********************************************************************
C     *                                                                  *
C     *       SUBROUTINE: ROWCHP                                         *
C     *       CALLED BY: MAIN                                            *
C     *       CALLING  ARGUMENTS: NONE                                   *
C     *       CALLED ROUTINES: IPHASE                                    *
C     *       OUTPUT FILES:                                              *
///   *                                                                  *
C     *                   9 - ODEQAUDDI.                                 *
C     *                   6 - STD PRINTER                                *
C     *       PURPOSE: DEFINES TYPE AND NAME OF EACH FLOW CONSERVATION   *
C     *                NODE CAPACITY,AND CONTROL OF INPUT CONSTRAINTS    *
C     *                (GOZO,CREQ,TREQ,TOTAUTH,UBSEG RESPECTIVELY)       *
C     *                AND THE CINC AND LINC FLOW CONTROL CONSTRAINTS    *
C     *                FOR ALL GRADES.                                   *
C     *       DATE: 15 APRIL 76                                          *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN     *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                *
C     *                8120 WOODMONT AVENUE                              *
C     *                BETHESDA,MARYLAND 20014                           *
C     *                                                                  *
C     ********************************************************************
      INCLUDE PROC1
C
C                      *****FORMAT STATEMENTS*****
C
10    FORMAT (1X,A1,2X,'N',J1,J2,A4)
20    FORMAT (1X,'E  N',J1,J2,A4)
40    FORMAT (1X,'E',2X,'WO',J2,'GOZO')
50    FORMAT (1X,'E',2X,'WO',J2,'LINC')
55    FORMAT (1X,'L',2X,'WO',J2,'UBSG')
57    FORMAT (1X,'L',2X,'TOTAUTH ')
C
C            *******GENERATE ROWS CHAPTER*******
      IF (JGRADE.LT.3) GO TO 80       @ SKIP WOM LINC,NOM TREQ,AND GOZO F
C      *** CONSTRAINTS FOR T-ZERO PRIME*****
        DO 60 J=1,NSPEC
        CALL IPHASE(J,$60)
C    ***************LINC FOR CAPTAINS ONLY *********
        IF (JGRADE.EQ.3) WRITE (9,50) NBRSPC(J)
C      ****GOZO****
        WRITE (9,40) NBRSPC(J)
      IROWCT(3) = IROWCT(3) + 1
      IF (JGRADE.EQ.3) IROWCT(4) = IROWCT(4)+ 1
60      CONTINUE
C       ***CONSTRAINTS FOR REQ AT TU***
        DO 70 J=1,NSPEC
        CALL IPHASE(J,$70)
C       *****TREQ*****
        ITYPE='L'
        WRITE (9,10) ITYPE,ZERO,NBRSPC(J),NAME(2)
      IROWCT(2) = IROWCT(2) + 1
70      CONTINUE
```

```
C          ••••••••••                 ••••••••••••         •••••••
80    IBEGIN=1
      JEND=NYRS
         DO 110 J=1,JEND
C          ••IEND=5 IS THE NUMBER  OF CONSTRAINTS FOR EA NODE
C          ••EXCEPT T=0,WHERE IEND =4,6 T(NYRS-1) WHERE CONST. 3-5
         IEND=5
         K=J-1
         IF (K.EQ.(NYRS-1)) IBEGIN=3
            DO 100 M=1,NSPEC
            CALL IPHASE(M,$100)
            K=J-1
            IF (K.EQ.0.AND.JGRADE.GT.3) IEND=4
            IF (K.LT.ICHG.AND.JGRADE.LT.3) IEND=4
            ITYPE='L'
               DO 90 L=IBEGIN,IEND
               K=J-1
               IF (JGRADE.EQ.6.AND.(L.EQ.1.OR.L.EQ.5)) GO TO 90
               IF (JGRADE.EQ.3.AND.(L.EQ.1.OR.(L.EQ.4.AND.J.GT.ICHG)))
     1          GO TO 90
               IF (JGRADE.EQ.2.AND.((L.EQ.1.AND.J.LE.ICHG).OR.(L.EQ.2.AN
     1D.J.EQ.NYRS))) GO TO 90
               IF (L.GT.2) ITYPE='E'
      IROWCT(L) = IROWCT(L) + 1
               IF (ITYPE.EQ.'L') K=J
               WRITE (9,10) ITYPE,K,NBRSPC(M),NAME(L)
90             CONTINUE
100         CONTINUE
110      CONTINUE
C       ••••CREQ CONSTRAINTS FOR T-N••••••
      ITYPE='L'
      IF (JGRADE.EQ.6.OR.JGRADE.EQ.3) GO TO 130
      IF (JGRADE.EQ.2.AND.ICHG.GE.NYRS) GO TO 130
         DO 120 M=1,NSPEC
         CALL IPHASE(M,$120)
         IROWCT(1) = IROWCT(1) + 1
C        ••••ICREQ••••
         WRITE (9,10) ITYPE,NYRS,NBRSPC(M),NAME(1)
120      CONTINUE
130   CONTINUE
C       •CONSTRAINTS FOR LAST PERIOD   GOZINTA=GOZOUTA
         DO 140 J=1,NSPEC
         CALL IPHASE(J,$140)
C        •••••GOZO••••
         WRITE (9,20) NYRS,NBRSPC(J),NAME(3)
      IROWCT(3) = IROWCT(3) + 1
C
140      CONTINUE
      IF(ISEG.NE.1) WRITE(9,57)
      IF (ISEG.GT.1 .OR. JGRADE.LT.4) RETURN
      DO 150 J = 1,NBRPRO
C         •••••••UBSG••••••
      WRITE(9,55)NPROB(J)
150   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROWOP
C     *****************************************************************
C     *****************************************************************
C     *                                                               *
C     *       SUBROUTINE: ROWOP                                       *
C     *       CALLED BY: MAIN (IMMEDIATE RETURN FOR CPTS AND BELOW)   *
C     *       CALLING  ARGUMENTS: NONE                                *
C     *       CALLED ROUTINES: IBITS,SET,IPHASE,JPHASE,OPT            *
C     *       OUTPUT FILES:                                           *
C     *                       9 - ODEQAUD01.                          *
C     *       PURPOSE: LOCATES REDUNDANT RTYPE CONSTRAINTS            *
C     *                WRITES  ROWCNT SUMMARY                         *
C     *                WRITES  RTYPE  CODES - OPTIONAL                *
C     *                WRITES  RTYPE  CONSTRAINT NAMES                *
C     *       DATE: 15 APRIL 76                                       *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN  *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *                8120 WOODMONT AVENUE                           *
C     *                BETHESDA,MARYLAND 20014                        *
C     *                                                               *
C     *****************************************************************
      INCLUDE PROC1
      INCLUDE PROC2
      INTEGER XLOS(50)
C     ************         *********         ***********     ******
C
C                          *****FORMAT STATEMENTS*****
C
10    FORMAT (1X,'E',2X,'R',J5)
30    FORMAT (1H1,'RESTRICTIVE FLOW CODES FOR GRADE',I3,' AT YEAR ',I3,
     * /,' CODES:  0 = NO FLOW  1 = FLOW(= CONSTRAINT) 2 =DELETE CODE',
     * ' 3 = UPPER BOUND ')
50    FORMAT (1H0,'SPECIALTY ',I2,2X,50I2)
C
C       ***********     ***********     ***********     ********
C       IDENTIFYS LAST ARC LEAVING A NODE, IS REDUNDANT TO RESTRICT FLOW
C
      IF (JGRADE.LT.4) RETURN
      IBGN=1
      IEND=NSPEC
        DO 80 J=1,NYRS
          DO 70 L=1,NSPEC
          CALL IPHASE(L,$65)
            DO 60 K=IBGN,IEND
            IF (IBITS(RTYPE,K).NE.1) GO TO 60
            J1=K
60          CONTINUE
          CALL SET (RTYPE,J1,2)
65        IBGN=IEND+1
          IEND=IBGN+(NSPEC-1)
70        CONTINUE
80      CONTINUE
C       ***********     ***********     ***********     ********
C
        DO 110 J=1,NYRS
```

```
      K=J-1
         DO 100 M=1,NSPEC
         CALL IPHASE(M,$100)
            DO 90 N=1,NSPEC
             CALL IPHASE(N,$90)
             CALL JPHASE(M,N,$90)
            INDEX=(K*NSPEC**2)+(M-1)*NSPEC+N
            IF (IBITS(RTYPE,INDEX).NE.1) GO TO 90
            LOC=(K*10000)+(NBRSPC(M)*100)+NBRSPC(N)
            WRITE (9,10) LOC
            IROWCT(6) = IROWCT(6) + 1
90          CONTINUE
100      CONTINUE
110   CONTINUE
C     **********      **********      **********      *******
      K1=1
      K2=NSPEC
      IF(IOPT(IRD).LT.1) RETURN                 @ REDUCED PRINT OUTPUT
        DO 140 J=1,NYRS
        K=J-1
        WRITE (6,30) JGRADE, K
         DO 130 M=1,NSPEC
         CALL IPHASE(M,$125)
            DO 120 KX=K1,K2
            XLOS(KX-K1+1)=IBITS(RTYPE,KX)
120         CONTINUE
         WRITE (6,50)NBRSPC(M),(XLOS(KX),KX=1,NSPEC)
125      K1=K2+1
         K2=K1+(NSPEC-1)
130      CONTINUE
140   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SORTW
C     ****************************************************************
C     *                                                              *
C     *       SUBROUTINE: SORTW                                      *
C     *       CALLED BY: MAIN                                        *
C     *       CALLING ARGUMENTS: NONE                                *
C     *       CALLED ROUTINES: SOPEN,SRREL,SSORT,SRRET               *
C     *       INPUT FILES: - 4 - ODSAPUD04.                          *
C     *                                                              *
C     *       OUTPUT FILES: - 4 - ODSAPUD04.                         *
C     *                                                              *
C     *       PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO SORT THE W- *
C     *                RECORDS FROM THE MATRIX GENERATOR CORRESPONDING *
C     *                TO THE W-RECORDS PRODUCED BY FMPS IN THE      *
C     *                SOLUTION FILE IN PREPARING FOR THE DATA BASE  *
C     *                CREATION.                                     *
C     *       DATE: 15 APRIL 76                                      *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                8120 WOODMONT AVENUE                          *
C     *                BETHESDA,MARYLAND 20014                       *
C     *                                                              *
C     ****************************************************************
       DIMENSION A(7),PARTBL(19)
       DATA PARTBL/1,1,12,0,0,1,
     *             2,1,12,0,0,3,
     *             3,1,12,0,0,2,99999/
      REWIND 4
       CALL SOPEN3($100,$300,7,3,PARTBL)
100    READ(4,901,END=200,ERR=200) A
901    FORMAT(3A2,4A6)
       CALL SRREL(A,7)
       GO TO 100
200    CALL SSORT
300    REWIND 4
301    CALL SRRET(A,N,$500)
       WRITE(4,901)A
       GO TO 301
500    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE SORTXY
C     ****************************************************************
C     *                                                              *
C     *     SUBROUTINE: SORTXY                                       *
C     *     CALLED BY: MAIN                                          *
C     *     CALLING ARGUMENTS: NONE                                  *
C     *     CALLED ROUTINES: SOPEN,SRREL,SSORT,SRRET                 *
C     *     INPUT FILES:                                             *
C     *                   - 8 - ODSAPUD08.                           *
C     *                   - 4 - ODSAPUD04.                           *
C     *                                                              *
C     *     OUTPUT FILES:                                            *
C     *                   - 4 - ODSAPUD04.                           *
C     *                                                              *
C     *     PURPOSE: THE PURPOSE OF THIS ROUTINE IS TO SORT THE XY-  *
C     *              RECORDS FROM THE MATRIX GENERATOR CORRESPONDING *
C     *              TO THE  XY- RECORDS PRODUCED BY FMPS IN THE     *
C     *              SOLUTION FILE AND ADD THEM TO THE ODSAPUD04     *
C     *              FILE IN PREPARING FOR THE DATA BASE CREATION.   *
C     *     DATE: 15 APRIL 76                                        *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *              8120 WOODMONT AVENUE                            *
C     *              BETHESDA,MARYLAND 20014                         *
C     *                                                              *
C     ****************************************************************
       DIMENSION A(8),PARTBL(19)
       DATA PARTBL/1,1,12,0,0,1,
     *             2,1,12,0,0,2,
     *             3,1,12,0,0,3,99999/
      REWIND 8
       CALL SOPEN3($100,$300,8,3,PARTBL)
100    READ(8,901,END=200,ERR=200) A
901    FORMAT(3A2,5A6)
       CALL SRREL(A,8)
       GO TO 100
200    CALL SSORT
300    CONTINUE
301    CALL SRRET(A,N,$500)
       WRITE(4,901)A
       GO TO 301
500    CONTINUE
       ENDFILE 4
       ENDFILE 4
       RETURN
       END
```

```
      FUNCTION VALID (ISPEC)
C     *************************************************************
C     *************************************************************
C     *                                                           *
C     *       FUNCTION  : VALID                                   *
C     *       CALLED BY:   RANGE,BOUNDS                           *
C     *       CALLING  ARGUMENTS: ISPEC -SPECIALTY TO BE CHECKED  *
C     *       CALLED ROUTINES: NONE                               *
C     *       PURPOSE:  TO DETERMINE IF ISPEC IS INCLUDED IN THE  *
C     *                 LIST OF ADVANCED ENTRY SPECIALTIES.       *
C     *                      VALID = 0.    ISPEC IS BES           *
C     *                      VALID = 1.    ISPEC IS AES           *
C     *       DATE: 15 APRIL 78                                   *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY         *
C     *                8120 WOODMONT AVENUE                       *
C     *                BETHESDA,MARYLAND 20014                    *
C     *                                                           *
C     *************************************************************
      INCLUDE PROC1
      VALID=0.
         DO 10 ICHECK=1,NBRAES
         IF (NBRSPC(ISPEC).NE.AES(ICHECK)) GO TO 10
         VALID=1.
         RETURN
10       CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COMBIN
C     ****************************************************************
C     *                                                              *
C     *         SUBROUTINE: COMBIN                                   *
C     *         CALLED BY: DATABASE                                  *
C     *         CALLING ARGUMENTS: NONE                              *
C     *         CALLED ROUTINES: NONE                                *
C     *         INPUT FILES:                                         *
C     *                        - 4 - ODSAPUD04.                      *
C     *                        - 10 - TEMPORARY FILE                 *
C     *         OUTPUT FILES:                                        *
C     *                        - 7 - ODSAPUD07.                      *
C     *         PURPOSE: MERGE THE TWO INPUT FILES TO PRODUCE THE INPUT
C     *                  FILE FOR THE CURRENT SEGMENT DATABASE.      *
C     *         DATE: 15 APRIL 78                                    *
C     *         AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN
C     *         COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY          *
C     *                  8120 WOODMONT AVENUE                        *
C     *                  BETHESDA,MARYLAND 20014                     *
C     *                                                              *
C     ****************************************************************
      DIMENSION IVAL(3),IN(10),IOUT(10),ISAVE(10)
      DATA IOUT/10*0/
C     READ FMPS PRODUCED DATA RECORD
1     READ(10,901,END=999)ITYPE,ALPHA,IYR,ISPEC1,ISPEC2,ISEG,IVAL,
     * INDEX
901   FORMAT(A1,A1,J1,2J2,J2,3J12,T2,I6)
      IF(ITYPE .NE.'R') GO TO 5
C     ROW RECORD = ROW RECORDS OUTPUT DIRECTLY TO FILE 7
C     NO FURTHER ACTION REQUIRED AT THIS POINT
40    WRITE(7,901)ITYPE,ALPHA,IYR,ISPEC1,ISPEC2,ISEG,IVAL
      GO TO 1
C
5     CONTINUE
      IF(ALPHA. NE. 'X') GO TO 20
C  ***** ROUTINE FOR X PREFIX COLUMNS
      IF(ISPEC1 .EQ. 0) GO TO  51
      IF (IYR .EQ. 9) GO TO  51
      IF(ISPEC1 .EQ. ISPEC2)GO TO  51
C     Y - RECORD DATA CHECK
20    IF (ALPHA.EQ.'Y' .AND. (ISPEC1.EQ.ISPEC2))GO TO 51
C     Y-M-M RECORDS WRITTEN OUT IMMEDIATELY
C     CHECK IF NEW RECORD ID EQUAL TO PREVIOUS RECORD ID
      IF (ISET.EQ.1) GO TO  23
C     READ MATRIX GENERATOR PRODUCED DATA RECORD
      READ(4,902,ERR=51,END=51)JINDEX,IN
902   FORMAT(A8,1X,4I1,I5,I4,I5,I4,I5,I4)
C          IS SOLUTION RECORD(INDEX) EQ MATRIX GENERATOR RECORD(JINDEX)
C          YES - GO CHECK WHETHER A X OR Y RECORD
C          NO - WRITE TO FILE 7,AND READ NEW SOLUTION RECORD
23    IF(INDEX.EQ.JINDEX) GO TO 999
C     SOLUTION RECORD DOES NOT MATCH M.G. RECORD
      ISET = 1
      GO TO 51
999   CONTINUE
```

```
      IF(ALPHA .NE.'Y') GO TO 320
C     Y - RECORD ENCOUNTERED - NEED ADDITIONAL DATA FROM ODSAPUD04.
      READ(4,902,ERR=51,END=51)KINDEX,ISAVE
      IF (KINDEX .NE. JINDEX) GO  TO 300
      IF ((ISAVE(7).GT.0) GO TO 310
      IF ((IN(5).GT.0) GO TO 315
      IN(5) = ISAVE(5)
      IN(6) = ISAVE(6)
      GO TO 320
310   IN(7) = ISAVE(7)
      IN(8) = ISAVE(8)
      GO TO 320
315   IN(9) = ISAVE(7)
      IN(10) = ISAVE(8)
      GO TO 320
C     JINDEX RECORD SHOULD BE WRITTEN OUT,BACK SPACE SO THAT
C     KINDEX RECORD CAN BE READ AGAIN.
300   BACKSPACE 4
320   ISET = 0
C     WRITE RECORD WHEN INDEX .EQ. JINDEX
      WRITE(7,907)ITYPE,ALPHA,IYR,ISPEC1,ISPEC2,ISEG,IVAL,IN
907   FORMAT(A1,A1,J1,2J2,J2,3J12,4J1,J5,J4,J5,J4,J5,J4)
      GO TO 1
C     WRITE RECORD WHEN INDEX .NE. JINDEX
51    WRITE(7,907)ITYPE,ALPHA,IYR,ISPEC1,ISPEC2,ISEG,IVAL,IOUT
      GO TO 1
9999  ENDFILE 7
      ENDFILE 7
      REWIND 7
      RETURN
      END
```

```
C
C     ******************************************************************
C     *                                                                *
C     *       PROGRAM: DATABASE                                        *
C     *       CALLED ROUTINES: NEWSAV,COMBIN,RECORD                    *
C     *       INPUT FILES:                                             *
C     *                 11.,ODDBSUDI.                                  *
C     *                 18.,ODSAPUD18.                                 *
C     *                  7.,ODSAPUD07.                                 *
C     *                  4.,ODSAPUD04.                                 *
C     *       OUTPUT FILE:                                             *
C     *                  3.,ODSAPUD03.                                 *
C     *                  7.,ODSAPUD07.                                 *
C     *       PURPOSE: THIS PROGRAM READS THE FMPS. FILE (ODDBSUDI)    *
C     *                AND EXTRACTS THE REQUIRED DATA FOR THE DATABASE, *
C     *                COMBINES IT WITH THE MATRIX GENERATOR DATA      *
C     *                FILE (ODSAPUD04.) AND PREPARES A SEGMENT DATA   *
C     *                BASE. IT THEN COMBINES THE SEGMENT DATA BASE    *
C     *                WITH THE CUMULATIVE DATA BASE AND PREPARES A    *
C     *                TEMPORARY UPDATED CUMULATIVE DATA BASE FOR      *
C     *                ANALYSIS (ODSAPUD03).                           *
C     *       DATE: 15 APRIL 76                                        *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *                8120 WOODMONT AVENUE                            *
C     *                BETHESDA,MARYLAND 20014                         *
C     *                                                                *
C     ******************************************************************
C
      CALL NEWSAV         @ THIS ROUTINE READS THE SOLUTION FILE
      CALL COMBIN         @ THIS ROUTINE COMBINES SOLUTION AND MATRIX DATA
      CALL RECORD          @ THIS ROUTINE GENERATES I-P RECORDS FOR DATA BASE
      STOP
      END
```

```
C     ***************************************************************
C     *                                                             *
C     *    PROGRAM    : DB-CORRECT                                  *
C     *    CALLED BY: NONE                                          *
C     *    CALLING ARGUMENTS: NONE                                  *
C     *    CALLED ROUTINES: ERTRAN                                  *
C     *    INPUT FILES: ODSAPUDOI                                   *
C     *               : 20.  TEMPORARY ROWS DATA                    *
C     *               : 21.  TEMPORARY COLUMNS DATA                 *
C     *                                                             *
C     *    OUTPUT FILES: 22: TEMPORARY ROWS AND COLUMNS DATA        *
C     *    PURPOSE: THIS PROGRAM ADDS THE ACTIVITY OF N(NYRS)--CREQ *
C     *             CONSTRAINTS TO THE ACTIVITY OF THE XN--         *
C     *             ARCS. FILE 22 IS NOT WRITTEN UPON WHEN GRADE    *
C     *             EQUAL TO 6 (CONDITION WORD IS SET, AND THEN     *
C     *             TESTED IN THE PFCAA.DB-MODIFY RUNSTREAM         *
C     *                                                             *
C     *    DATE: 15 APRIL 76                                        *
C     *    AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *    COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *             8120 WOODMONT AVENUE                            *
C     *             BETHESDA,MARYLAND 20014                         *
C     *                                                             *
C     ***************************************************************
      INTEGER A(8)
      INTEGER SAVE(100)
      READ(15,900)NSPEC,NYRS,JGRADE
900   FORMAT(I2,2X,I2,17X,I1)
      IF (JGRADE.EQ.8) GO TO 200
C      ***READ ROW RECORDS****
1     READ(20,910,END=100)ID,A
910   FORMAT(A2,J1,J2,J2,J2,3J12,J4)
C      ***TEST FOR N(NYRS)--CREQ RECORD***
      IF(A(1).NE.NYRS)WRITE(22,910)ID,A
      INDEX=A(2)
      IF(A(1).EQ.NYRS)SAVE(INDEX)=A(5)
C      ***SAVE ACTIVITY VALUE OF CREQ RECORD***
      GO TO 1
C      ***READ COLUMN RECORDS***
100   READ(21,910,END=1000)ID,A
      IF(A(1).NE.9)WRITE(22,910)ID,A
      IF(A(1).NE.9)GO TO 100
      IGRADE=A(4)/10.
      IF(IGRADE.LE.JGRADE)WRITE(22,910)ID,A
      IF(IGRADE.LE.JGRADE)GO TO 100
      INDEX=A(2)
C        ***ADD ACTIVITY OF CREQ RECORD TO Z9(XN--)RECORD***
      A(5)=A(5)+SAVE(INDEX)
C           ***INSURE THAT ACTIVITY .LE. UL***
      IF(A(5).GT.A(7))A(5)=A(7)
      WRITE(22,910)ID,A
      GO TO 100
C     ****@SETC  6/T3 ******
200     CALL ERTRAN(7,JGRADE)
1000  ENDFILE 22
      ENDFILE 22
      REWIND 22
      STOP
      END
```


BEST AVAILABLE COPY

```
C     ......................................................................
C     .                                                                    .
C     .        PROGRAM    : DBGEN                                          .
C     .        CALLED BY: NONE                                             .
C     .        CALLING ARGUMENTS: NONE                                     .
C     .        CALLED ROUTINES: OPENS,WRITES,CLOSEM (MIRADS)               .
C     .        INPUT FILES: - 9 - ODSAPUD07 OR ODSAPUD03                   .
C     .        OUTPUT FILES: MASCUMSEG                                     .
C     .                      MASCURSEG                                     .
C     .        PURPOSE: THIS PROGRAM READS THE CARD IMAGE                  .
C     .                 DATA BASE INPUT FILES                              .
C     .                 AND CREATES THE MIRADS DATA BASE FILE              .
C     .                 MASCURSEG OR MASCUMSEG.                            .
C     .        DATE: 15 APRIL 76                                           .
C     .        AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN      .
C     .        COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                 .
C     .                 8120 WOODMONT AVENUE                               .
C     .                 BETHESDA,MARYLAND 20014                            .
C     .                                                                    .
C     ......................................................................
      DIMENSION IBUF(1807),IFILE(2),ILOC(14)
      DATA IFILE/'MAS   ','      '/
      IRECSZ=14
      IBLKSZ=128
      IRECNO=1
      CALL OPENS('MAS         ',IRECSZ,IBLKSZ,IBUF)
    5 READ(9,20,END=10) (ILOC(I),I=1,14)
   20 FORMAT(13A6,A2)
      CALL WRITES('MAS         ',IRECNO,ILOC)
      IRECNO=IRECNO+1
      GO TO 5
   10 WRITE(6,15) IRECNO
   15 FORMAT(5X,16HRECORDS STORED =,I5)
      CALL CLOSEM(IFILE)
      STOP
      END
```

```
C     ****************************************************************
C     *                                                              *
C     *       PROGRAM: HITFILE/INTERFACE                             *
C     *       CALLED BY: NONE                                        *
C     *       CALLING ARGUMENTS: NONE                                *
C     *       CALLED ROUTINES: ERTRAN, OPENS, READS, CLOSEM          *
C     *       INPUT  FILES: HITFILE                                  *
C     *                     MASCUMSEG                                *
C     *       OUTPUT FILES: 9 - TEMPORARY FILE                       *
C     *       PURPOSE: THIS PROGRAM READS THE MIRADS HITFILE         *
C     *                TO OBTAIN THE INDICES TO RECORDS IN MASCUMSEG *
C     *                THEN READS THE APPROPRIATE RECORDS IN         *
C     *                MASCUMSEG AND WRITES THE RECORD TO FILE 9./   *
C     *       DATE: 15 APRIL 76                                      *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                8120 WOODMONT AVENUE                          *
C     *                BETHESDA,MARYLAND 20014                       *
C     *                                                              *
C     ****************************************************************
      DIMENSION IBUF(435),IREC(3)
      DIMENSION JBUF(2000),JREC(896)
      DIMENSION IFILE(2),IASG1(3),IASG3(5)
      DATA IASG1/'@ASG,T    9.    .  '/
      DATA IASG3/'@ASG,A                          .   '/
160    FORMAT(13A6,A2)
310   FORMAT(' OUTPUT IS NOW IN TEMPORARY FILE  9',/
     * ' PLEASE COPY IT TO YOUR FILE')
      CALL ERTRAN(6,IASG1)
C     **USE HITFILE TO OBTAIN INDICES FOR MASCUMSEG**
      CALL OPENS('HITFILE      ',3,140,IBUF)
      IFILE(1)= 'MASCUM'
      IFILE(2)='SEG   '
      IASG3(3)=IFILE(1)
      IASG3(4)=IFILE(2)
      CALL ERTRAN(6,IASG3)
      KWRDS=14
      KBLKS=1792/KWRDS
      CALL OPENS(IFILE,KWRDS,KBLKS,JBUF)
      IA=0
5     IA=IA+1
C     **READ THE HITFILE**
      CALL READS('HITFILE      ',IA,IREC,ISW)
C     **TEST FOR LAST RECORD**
      IF(ISW.EQ.1)GO TO 50
      JA=IREC(1)
C     **READ MASCUMSEG**
      CALL READS(IFILE,JA,JREC,ISW)
      WRITE( 9,160) (JREC(M),M=1,14)
      GO TO 5
50    CALL CLOSE ('HITFILE      ')
      CALL CLOSE (IFILE)
      ENDFILE  9
      WRITE(6,310)
      STOP
      END
```

```
      SUBROUTINE MASKCK(ALPHA,ALPHA1,BETA,S,JGRADE)
C     *****************************************************************
C     *                                                               *
C     *        SUBROUTINE: MASKCK                                     *
C     *        CALLED BY: NEWSAV                                      *
C     *        CALLING ARGUMENTS; ALPHA,ALPHA1,BETA,JGRADE            *
C     *        CALLED ROUTINES: NONE                                  *
C     *        INPUT FILES: NONE                                      *
C     *                                                               *
C     *        OUTPUT FILES: NONE                                     *
C     *                                                               *
C     *        PURPOSE: CHECKS ALL RECORDS FROM ODDBSUD1, SO THAT     *
C     *                 GOZO, LINC, CINC, R-TYPE CONSTRAINTS FOR      *
C     *                 ALL GRADES CAN BE EXCLUDED FROM THE DATA      *
C     *                 BASE. ADDITIONALLY, CHECKS FOR RES TYPE       *
C     *                 CONSTRAINTS IN FIELD GRADE SEGMENTS SO        *
C     *                 THEY MAY BE EXCLUDED                          *
C     *        DATE: 15 APRIL 76                                      *
C     *        AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *        COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY            *
C     *                 8120 WOODMONT AVENUE                          *
C     *                 BETHESDA,MARYLAND 20014                       *
C     *                                                               *
C     *****************************************************************
C        ***MASK OUT GOZO,LINC &CINC CONSTRAINTS**
         IF(BETA .EQ.'GOZO' .OR. BETA.EQ.'LINC' .OR. BETA .EQ. 'CINC')
     A RETURN 4
C        ***FOR ALL GRADES,MASK OUT R----- CONSTRAINTS**
         IF(ALPHA1 .EQ. 'R')RETURN 4
C       ***FOR FIELD GRADES, MASK OUT RES----- CONSTRAINTS**
         IF(JGRADE .GT.3 .AND.ALPHA .EQ. 'RES')RETURN 4
         RETURN
         END
```

```
      SUBROUTINE NEWSAV
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *     SUBROUTINE: NEWSAV                                       *
C     *     CALLED BY: DATABASE                                      *
C     *     CALLED ROUTINES: NONE                                    *
C     *     INPUT FILES:                                             *
C     *              5 - CARD READER I/P OF GRADE AND SEGMENT ID     *
C     *              11 - ODDBSUD1.                                  *
C     *     OUTPUT FILES:                                            *
C     *              10 - TEMPORARY                                  *
C     *     PURPOSE: THE PURPOSE OF THIS PROGRAM IS TO CREATE A CARD *
C     *              IMAGE FILE FROM THE FMPS O/P FILE,THAT CAN BE   *
C     *              COMBINED WITH SELECTED MATRIX GENERATOR OUTPUT  *
C     *              (FILE ODSAPUD04)TO PRODUCE AN INPUT DATABASE    *
C     *              FILE FOR MIRADS. PFODSAP.COMBIN MERGES THE TWO  *
C     *              FILES TO PRODUCE ODSAPUD07.                     *
C     *     DATE: 15 APRIL 76                                        *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C     *              8120 WOODMONT AVENUE                            *
C     *              BETHESDA,MARYLAND 20014                         *
C     *                                                              *
C     ****************************************************************
      DIMENSION L(19),IHDR(6),IVAL(3),NEXT(456),NAME(2),ISAVE(3)
      DATA Z/0/
C
C     L IS USED FOR RECORD SEPARATION WITHIN BLOCKS
C     HEADER IS USED TO STORE BLOCK INFORMATION
C     IVAL CONTAINS DECODED NUMERICAL INFORMATION
C     NAME IS THE ROW OR COLUMN RECORD NAME
C     ISAVE IS USED IN CREQ AND TREQ RECORDS
C     SOLFIL IS THE SOLUTION FILE
C     NEXT IS EQUIVALENCED TO SOLFIL
C     IHR IS EQUIVALENCED TO HDR (FIRST 3-DP WORDS IN RECORD)
C
      DATA L/15,27,39,51,6,18,30,42,54,9,21,33,45,57,12,24,36,48,60/
      DOUBLE PRECISION SOLFIL(228),HDR(3)
      EQUIVALENCE (SOLFIL(1),NEXT(1)),(HDR(1),IHDR(1))
905      FORMAT(A1)
907   FORMAT(2I1)
      READ(11)HDR
      READ(11)HDR
      READ(5,907) JGRADE,ISEG
      DO 10  I = 1,4
      IBGN = (I-1)*57 + 1
      IEND = IBGN + 56
      READ(11)HDR,(SOLFIL(K),K=IBGN,IEND)
C        **ITYPE IS EQUAL TO ROWS OR COLU**
      ITYPE = IHDR(3)
C        **CHECK FOR LAST ROW OR COLUMN RECORD,IHDR(5)=RN**
C           IF RN IS NEGATIVE,THEN NOT LAST RECORD***
      IF (IHDR(5).GT.0)GO TO 997
10    CONTINUE
```

```
      NREC = 19
12    DO 20 I = 1,NREC
      IGRADE = JGRADE
      ISEGX  = ISEG
      KSPEC=0
      J = 12*(I-1)
      NAME(1) = NEXT(2*J + 1)
         ALPHA = 'A'
         ALPHA1 = 'A'
         BETA = 'A'
C        ..IMASK OUT OBJECTIVE F(X)..
         DECODE(4,918,NAME(1))OBJECT
918        FORMAT(A4)
          IF(OBJECT.EQ. 'OBJE')GO TO 20
C         ...MASK OUT UR CONSTRAINTS..
       DECODE(4,917,NAME(1)) UR,NBR
917        FORMAT(2A2)
        IF(UR .EQ. 'UR') GO TO 20
C          ..DETERMINE RES OR TOT DATA..
      DECODE(4,910,NAME(1))ALPHA,DELTA
910   FORMAT (A3,A1)
      IF(ALPHA.NE.'RES' .AND.ALPHA .NE.'TOT')
     A DECODE(4,903,NAME(1))ALPHA1,IYR,ISPEC
      NAME(2) = NEXT(2*J + 2)
C          ..GET LAST FOUR CHARACTERS OF NAME..
      DECODE(4,911,NAME(2))BETA
911   FORMAT (A4)
            CALL MASKCK(ALPHA,ALPHA1,BETA,$20,JGRADE)
C         ....RES----- RECORD....
      IF(ALPHA.EQ.'RES')GO TO 21
840    IF(ALPHA .NE. 'TOT') GO TO 845
C         ...TOTAUTH RECORD....
        DECODE(1,905,NAME(1)) ALPHA
        IYR='0'
       ISPEC=0
       KEY=0
       GO TO 850
845   DECODE(4,903,NAME(1))ALPHA,IYR,ISPEC
903   FORMAT (A1,A1,I2)
      NAME(2) = NEXT(2*J + 2)
      KEY    = FLD(0,6,NAME(2))
      IF (ALPHA .NE.'N') GO TO 800
C
C     ROW RECORD (DETERMINE CREQ OR TREQ)
C
      IF (KEY .NE. '@@@@@C')GO TO 850
      IGRADE = JGRADE + 1
      ISEGX  = ISEG +2
      GO TO 850
C         ...WO--UBSG RECORD IF EQUAL...
800     IF(KEY .EQ. '@@@@@U') GO TO 850
C         ...COLUMN RECORD X,Y,W OR XN.....
825   DECODE(2,904,NAME(2))KSPEC
904   FORMAT(I2)
      ISEGX = 1
      IF (ALPHA.EQ.'X'.AND. ISPEC.EQ.KSPEC) ISEGX=ISEG
      IF ((IYR .EQ.'N')IYR = '9'
```

```
      IF (ALPHA.NE.'Y') GO TO 850
      IGRADE = JGRADE+1
      ISEGX = 3
      IF(NSPEC.NE.KSPEC) ISEGX = ISEG + 2
      GO TO 850
850   DO 25 K = 1,3
      KKKKK= (J+K*3)*2 -1
      IF (NEXT(KKKKK).EQ.'NONE') SOLFIL(J+K*3) = 0
      IVAL(K) = SOLFIL(J+K*3) + .5   @ ROUND OFF RULE
      IF (KEY .EQ. '@@@@@C') ISAVE(K) = IVAL(K)
25    CONTINUE
         IF(KEY.NE.'@@@@@T') GO TO 28
C        TREQ RECORD
C        SAVE DIFFERENCE BETWEEN ORIGINAL UL AND ACTIVITY OF TREQ RECORD
         IVAL(2)=IVAL(3) - IVAL(1)
C        SET TREQ UL TO TREQ UL - CREQ UL
         IVAL(3)=IVAL(3) - ISAVE(3)
C        SUBTRACT CREQ ACTIVITY FROM TREQ ACTIVITY
         IVAL(1)=IVAL(1)-ISAVE(1)
28    KEY = 0
      IF(ALPHA.EQ.'RES')GO TO 21
      WRITE (10,902) IHDR(3),ALPHA,IYR,ISPEC,KSPEC,IGRADE,ISEGX,IVAL
902   FORMAT (2A1,A1,2J2,2I1,3J12)
      GO TO 20
C        **CHANGE MINUS ACTIVITY VALUE TO PLUS FOR RES----**
21    IVAL(1) = ABS(SOLFIL(J+3)) + 0.5
      WRITE (10,912) IHDR(3),DELTA,BETA,IGRADE,ISEGX,IVAL(1),Z,Z
912   FORMAT (A1,'V',A1,A4,2I1,3J12)
20    CONTINUE
      GO TO 1
997         DO 30 NREC = 1,19
C       **IHDR(6) IS LAST WORD USED IN RECORD FOR DATA**
      IF (IHDR(6) .EQ. L(NREC)) GO TO 12
30    CONTINUE
      ENDFILE 10
      ENDFILE 10
      REWIND 10
      RETURN
      END
```

```
C     ***************************************************************
C     *                                                             *
C     *     PROGRAM.       PROCUREMENT                              *
C     *     CALLED BY:     NONE                                     *
C     *     INPUT FILES:   12 - ODPOPUD01                           *
C     *                    15 - ODD01UD9D                           *
C     *                                                             *
C     *     OUTPUT FILES:  6 - PRINTER                              *
C     *                    7 - ALT PRINT FILE                       *
C     *                                                             *
C     *     PURPOSE:       THIS PROGRAM PRODUCES THE ODSAS          *
C     *                    PROCUREMENT REPORT                       *
C     *                                                             *
C     *     DATE:          15 APRIL 76                              *
C     *     AUTHORS: MAJ J.D. THOMAS. MAJ J.W. OLSON. MR R.L. BROWN *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY             *
C     *              8120 WOODMONT AVENUE                           *
C     *              BETHESDA. MARYLAND  20014                      *
C     *                                                             *
C     ***************************************************************
      INTEGER ABSA,YEAR,FROM,TO,GRD,ACT
      INTEGER TOTACT(100),TOTLCA(100),REQ(100)
      INTEGER TOTLAS(100)
      PARAMETER NBRSPC=45
      PARAMETER NBRCBT=4
      PARAMETER NBRAES=16
      PARAMETER NBRBES=NBRSPC -NBRAES
      PARAMETER NAESPL=NBRAES+2
      INTEGER BESNBR(NBRBES)
      DIMENSION PERCA(NBRAES,100),PERCB(NBRAES,100)
      DIMENSION RATE(7,31)
      DIMENSION ALLBES(100,NAESPL),BESMIN(100,NAESPL)
      INTEGER A(100,100)
     INTEGER AESNBR(NBRAES),SPCNBR(NBRSPC),CBTNBR(NBRCBT)
      DIMENSION TOT1(22),TOT2(22)
      DATA AESNBR/15,41,45,46,47,48,49,51,52,53,54,86,91,93,95,97/
    DATA SPCNBR/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
   145,46,47,48,49,51,52,53,54,71,72,73,74,75,76,77,81,82,83,86,87,88,
   291,92,93,95,97/
      DATA CBTNBR/11,12,13,14/
    DATA BESNBR/11,12,13,14,21,25,26,27,28,31,35,36,37,42,43,44,
   171,72,73,74,75,76,77,81,82,83,87,88,92/
900   FORMAT(2A1,I1,3J2,2J12)
901   FORMAT('OSPECIALTY NUMBER',I3/1H0,5X,'PAIRED WITH  ',/
   *'   SPECIALTIES:',
   *4X,'QNTY',4X,'PCNT',4X,'QNTY',4X,'PCNT',4X,'QNTY',4X,
   *'PCNT'/27X,'ALL ',12X,'-AES',12X,'-CBT&AES')
902   FORMAT(1H0,10X,I3,3X,I5,3X,F5.1)
903   FORMAT(1H0,5X,'TOTALS',5X,I5,12X,F5.0,11X,F5.0)
904   FORMAT('0SPECIALTY  T-0 POP  LT PRMTS',5X,'LTS',10X,'CPTS',7X,
   *'TOT ATTR  ','  U/F REQ  TOT REQ')
910   FORMAT(1H1,24X,'O D S A S    P R O C U R E M E N T    ',
   *'R E P O R T',
   *'    P A R T',3X,I2,' O F    2')
911   FORMAT(1H0,37X,'INCLUDES PRORATION OF AES TO ALL BASIC ENTRY',
   *' SPECIALTIES')
913   FORMAT(1H0,37X,'EXCLUDES PRORATION OF AES TO ALL COMBAT ARMS',
   *' SPECIALTIES')
912   FORMAT('OSPECIALTY',10X,'BES',5X,'*******',30X,
   *'PRORATED AES REQUIREMENTS',29X,'*****',5X,'TOTAL'/
```

```
     *'ONUMBER',13X,'REQMT',16(3X,I3),2X,'REQMT')
920       FORMAT(1H0,3X,I3,13X,F5.0,3X,16(F4.0,2X),F5.0)
930       FORMAT(1H0,3X,I3,13X,F5.0,98X,F5.0)
940       FORMAT(1H0,3X,I3,4X,I8,2X,F8.4,2X,F10.4,2X,F10.4,2X,
     *F10.4,I8,F10.4)
945       FORMAT(1H0,10X,'TOTAL',2X,F7.0,2X,16(F5.0,1X),F7.0)
950       FORMAT(1H0,10X,I3,3X,I5,3X,F5.1,3X,I5,3X,F5.1)
955       FORMAT(1H0,10X,I3,3X,I5,3X,F5.1,3X,I5,3X,F5.1,3X,I5,3X,F5.1)
959       FORMAT(1H0,'DO YOU WANT ANY OPTIONAL REPORTS - ANSWER',
     *' YES OR NO')
960       FORMAT(1H0,'**REPORTS ARE IN FILE 9OALTPRNT**',/1H0,
     *'TO PRINT THE REPORTS - TYPE IN @SYM,U 9OALTPRNT.,,PR',/1H0,
     *'USE THE ED PROCESSOR TO VIEW FROM THE TERMINAL',
     *' I.E. TYPE IN @ED,R 9OALTPRNT.')
961       FORMAT(A3)
C         **ESTABLISH LUN 7 AS ALTERNATE PRINT FILE**
C         CALL NTAB(1,1,1,1,1,6,1,5,0,7)
          WRITE(7,940)
          READ(12) RATE
1         READ(15,900,END=5000)ID,PREFIX,YEAR,FROM,TO,GRD,ACT,LL
          IF(ID.NE.'R')GO TO 10
          IF(PREFIX.EQ.'V')GO TO 10
          REQ(FROM)=ACT
          Y=REQ(FROM)*RATE(2,19)
          X=FLOAT(REQ(FROM)) - Y
C         **COMPUTE TOTAL ATTRITION AND ADD TO UNFILLED REQUIREMENTS
          TOTATR=(Y * RATE(2,1)) + (X * RATE(2,14))
          REQ(FROM)=TOTATR + LL + 0.5
          WRITE(7,940)FROM,ACT,RATE(2,19),Y,X,TOTATR,LL,REQ(FROM)
          GO TO 1
10        A(FROM,TO)=A(FROM,TO)+ACT
          A(TO,FROM)=A(TO,FROM)+ACT
          GO TO 1
C         **COMPUTE TOTAL ACTIVITY FOR EACH SPECIALTY**
5000      DO 5039 JJ=1,NBRSPC
          J=SPCNBR(JJ)
          DO 5020 K=1,100
          TOTACT(J)=TOTACT(J)+A(J,K)
5020      CONTINUE
5039      CONTINUE
C         **COMPUTE PORTION OF TOTAL ACTIVITY DUE TO CBT ARMS & AES/AES
          DO 5040 JJ=1,NBRSPC
          J=SPCNBR(JJ)
          DO 5030 L=1,NBRAES
          IF(J.NE.AESNBR(L))GO TO 5030
C         **J=AES
          DO 5025 K=1,100
          DO 5023 M=1,NBRCBT
          IF (K.NE.CBTNBR(M))GO TO 5023
          TOTLCA(J)=TOTLCA(J)+A(J,K)
          GO TO 5025
5023      CONTINUE
          DO 5024 M=1,NBRAES
          IF(K.NE.AESNBR(M))GO TO 5024
          TOTLAS(J)=TOTLAS(J)+A(J,K)
          GO TO 5025
5024      CONTINUE
5025      CONTINUE
5030      CONTINUE
5040      CONTINUE
```

```
C         ••COMPUTE AND PRINT PERCENTAGES FOR EACH SPECIALTY••
          DO 5500 JJ=1,NBRSPC
          J=SPCNBR(JJ)
          WRITE(7,901) J
5161      ALLBES(J,1)=REQ(J)
          ALLBES(J,NBRAES+2)=ALLBES(J,NBRAES+2) + REQ(J)
          TOT1(NAESPL)=TOT1(NAESPL) + REQ(J)
C         ••FILL BESMIN ARRAY FOR NON CBT ARMS
          CBTCHK=0.0
          DO 5100 N=1,NBRCBT
          IF(J.EQ.CBTNBR(N))CBTCHK=1.0
5100      CONTINUE
          BESMIN(J,1)=REQ(J)
          BESMIN(J,NBRAES+2)=BESMIN(J,NBRAES+2) + REQ(J)
          TOT2(NAESPL)=TOT2(NAESPL) + REQ(J)
C         COMPUTE PERCENT PARTICIPATION IN SPEC J
5162      DO 5400 K=1,NBRSPC
          INDEX=SPCNBR(K)
          IF(A(J,INDEX).EQ.0)GO TO 5400
          ABSA= A(J,INDEX)
          PERCNT=FLOAT(ABSA)/FLOAT(TOTACT(J)) *100.0
C         COMPUTE ALT % PARTICIPATION IN AES BY EXCL CBT ARMS
C            AND/OR AES PAIRS
          AESCHK=0.0
          DO 5190 L=1,NBRAES
          IF(J.NE.AESNBR(L))GO TO 5190
C         J IS AN AES
          JJJ=L
          AESCHK=1.0
          DO 5195 LL=1,NBRAES
          IF(INDEX.NE.AESNBR(LL))GO TO 5195
C         ••J AND K ARE BOTH AES
          WRITE(7,902)INDEX,ABSA,PERCNT
          GO TO 5400
5195      CONTINUE
          DENOM=FLOAT(TOTACT(J)-TOTLAS(J))
          PERCB(JJJ,INDEX)=FLOAT(ABSA)/DENOM  *100.0
          DO 5197 LL=1,NBRCBT
          IF(INDEX.NE.CBTNBR(LL))GO TO 5197
C         ••J=AES AND K=CBT ARM
5200      WRITE(7,950)INDEX,ABSA,PERCNT,ABSA,PERCB(JJJ,INDEX)
          GO TO 5400
5197      CONTINUE
C         ••J=AES AND K NE CBT OR AES
          DENOMA=FLOAT(TOTACT(J)-TOTLAS(J)-TOTLCA(J))
          PERCA(JJJ,INDEX)=FLOAT(ABSA)/DENOMA * 100.0
5201     WRITE(7,955)INDEX,ABSA,PERCNT,ABSA,PERCB(JJJ,INDEX),ABSA,
     *PERCA(JJJ,INDEX)
          GO TO 5400
5190      CONTINUE
          WRITE(7,902)INDEX,ABSA,PERCNT
5400      CONTINUE
          IF(CBTCHK.EQ.1.0)WRITE(7,903)TOTACT(J)
          IF(CBTCHK.EQ.1.0)GO TO 5500
          WRITE(7,903)TOTACT(J),DENOM,DENOMA
5500      CONTINUE
C         ••FILL REMAINDER OF ALLBES AND BESMIN ARRAYS
```

```
      DO 5900 JJ=1,NBRSPC
      J=SPCNBR(JJ)
      DO 5899 M=1,NBRBES
      IF(J.EQ.BESNBR(M))GO TO 5550
5899  CONTINUE
      GO TO 5900
C     **J=BES
5550  TOT1(1)=TOT1(1)+REQ(J)
      TOT2(1)=TOT2(1)+REQ(J)
      DO 5750 KK=1,NBRAES
      K=AESNBR(KK)
      MM=KK+1
      ALLBES(J,MM)=FLOAT(REQ(K))*(PERCB(KK,J)/100.0)
      ALLBES(J,NAESPL)=ALLBES(J,NAESPL)+ALLBES(J,MM)
      TOT1(MM)=TOT1(MM)+ALLBES(J,MM)
C     TOT1(NBRAES+1)=TOT1(NBRAES+1)+ALLBES(J,MM)
C     **TEST IF J IS A CBT ARM
      DO 5700 N=1,NBRCBT
      IF(J.EQ.CBTNBR(N))GO TO 5750
5700  CONTINUE
C     **J=CBT ARM
      BESMIN(J,MM)=FLOAT(REQ(K))*(PERCA(KK,J)/100.0)
      BESMIN(J,NAESPL)=BESMIN(J,NAESPL)+BESMIN(J,MM)
      TOT2(MM)=TOT2(MM)+BESMIN(J,MM)
C     TOT2(NBRAES+1)=TOT2(NBRAES+1)+BESMIN(J,MM)
5750  CONTINUE
5900  CONTINUE
C     **WRITE PART 1 OF PROCUREMENT REPORT**
      KNT=1
      WRITE(7,910)KNT
      WRITE(7,911)
      WRITE(7,912) (AESNBR(K),K=1,NBRAES)
      DO 6000 L=1,NBRSPC
      J=SPCNBR(L)
      DO 5800 N=1,NBRBES
      IF(J.NE.BESNBR(N))GO TO 5800
      WRITE(7,920) J,(ALLBES(J,K),K=1,NAESPL)
5800  CONTINUE
6000  CONTINUE
      IEND=NBRAES+1
      WRITE(7,945) (TOT1(J),J=1,NAESPL)
C     **WRITE PART 2 OF PROCUREMENT REPORT**
      KNT=2
      WRITE(7,910) KNT
      WRITE(7,913)
      WRITE(7,912)  (AESNBR(K),K=1,NBRAES)
      DO 7000 L=1,NBRSPC
      J=SPCNBR(L)
      DO 6800 N=1,NBRBES
      IF(J.NE.BESNBR(N))GO TO 6800
      DO 6900 M=1,NBRCBT
      IF(J.NE.CBTNBR(M)) GO TO 6900
      WRITE(7,930) J,BESMIN(J,1),BESMIN(J,NAESPL)
      GO TO 7000
6900  CONTINUE
      WRITE(7,920)  J,(BESMIN(J,K),K=1,NAESPL)
      GO TO 7000
```

```
6800      CONTINUE
7000      CONTINUE
          WRITE(7,945) (TOT2(J),J=1,NAESPL)
          WRITE(6,959)
          READ(5,961) OPTION
          IF(OPTION.EQ.'YES')CALL OPTRPT
          WRITE(6,960)
          STOP
C
          SUBROUTINE OPTRPT
          INTEGER OPTNBR,SPNBR
899       FORMAT(1H1,24X,'O D S A S    P R O C U R E M E N T    ',
     *'R E P O R T')
900       FORMAT()
905       FORMAT(1H0,37X,'***O P T I O N',I3,' ***')
910       FORMAT(1H0,'TYPE OPTION NUMBER COMMA AES NBR',/1H0,
     *'OPTION 1 - PRORATE THE IDENTIFIED SPECIALTY TO ALL AES',
     */1H0,'OPTIONS 2-9 ARE UNDEFINED AT THIS TIME')
912       FORMAT('0SPECIALTY',10X,'BES',5X,'*******',30X,
     *'PRORATED AES REQUIREMENTS',29X,'****',5X,'TOTAL'/
     *'0NUMBER',13X,'REQMT',16(3X,I3),2X,'REQMT')
915     FORMAT(1H0,'INVALID AES NBR ENTERED - I-P WAS ',I3,' TRY AGAIN')
920       FORMAT(1H0,3X,I3,13X,F5.0,3X,16(F4.0,2X),F5.0)
930   .   FORMAT(1H0,3X,I3,13X,F5.0,98X,F5.0)
945       FORMAT(1H0,10X,'TOTAL',2X,F7.0,2X,16(F5.0,1X),F7.0)
950     FORMAT(1H0,'IF NO FURTHER OPTIONS DESIRED - TYPE THE FOLLOWING',
     */1H0,'10,4')
955       FORMAT(1H0,'ENTER OPTION 1,AES NUMBER IF ANY OTHER AES ARE',
     *' TO BE PRORATED')
          WRITE(7,899)
          KNT=0
          NBR=1
1         WRITE(6,910)
          WRITE(6,950)
          READ(5,900)OPTNBR,SPNBR
          IF(KNT.EQ.0 .AND. OPTNBR.NE.NBR)GO TO 2
          KNT=KNT+1
          IF(OPTNBR.NE.NBR)GO TO 1000
2         DO 10 J=1,NBRAES
          IF(SPNBR.EQ.AESNBR(J))GO TO 20
10        CONTINUE
          WRITE(7,915)SPNBR
           GO TO 1
20        GO TO (300,400,500,500,500,500,500,500,500),OPTNBR
C         **OPTION ONE**
300       CONTINUE
          NBR=OPTNBR
          WRITE(6,955)
          DO 310 K=1,100
          DIFF=ALLBES(K,J+1) - BESMIN(K,J+1)
          BESMIN(K,J+1)=ALLBES(K,J+1)
          BESMIN(K,NAESPL)=BESMIN(K,NAESPL) + DIFF
310       CONTINUE
          GO TO 1
C         **OPTION TWO**
400       CONTINUE
          NBR=OPTNBR
```

```
      GO TO 1
C     **OPTION THREE**
500   CONTINUE
      NBR=OPTNBR
      GO TO 1
C     **WRITE BESMIN ARRAY**
1000  WRITE(7,905)NBR
      WRITE(7,912) (AESNBR(K),K=1,NBRAES)
      DO 7000 L=1,NBRSPC
      J=SPCNBR(L)
      DO 6800 N=1,NBRBES
      IF(J.NE.BESNBR(N))GO TO 6800
      DO 6900 M=1,NBRCBT
      IF(J.NE.CBTNBR(M)) GO TO 6900
      WRITE(7,920)J,(BESMIN(J,K),K=1,NAESPL)
      GO TO 7000
6900  CONTINUE
      WRITE(7,920)  J,(BESMIN(J,K),K=1,NAESPL)
      GO TO 7000
6800  CONTINUE
7000  CONTINUE
      WRITE(7,945) (TOT2(J),J=1,NAESPL)
      IF(OPTNBR.GT.9 .OR. OPTNBR .LT.1)GO TO 9000
      GO TO 2
9000  RETURN
      END
```

```
      SUBROUTINE RECORD
C
C     ***************************************************************
C     *                                                             *
C     *       SUBROUTINE: RECORD                                    *
C     *       CALLED BY: DATABASE                                   *
C     *       CALLING ARGUMENTS: NONE                               *
C     *       CALLED ROUTINES: SOPEN3,SRREL,SSORT,SSRET             *
C     *       INPUT FILES: ODSAPUD07,,ODSAPUD18,                    *
C     *       OUTPUT FILES: ODSAPUD03,                              *
C     *       PURPOSE:THIS PROGRAM CREATES THE INPUT RECORDS FOR    *
C     *               THE CUMULATIVE DATABASE BY ADDING THE CURSEG  *
C     *               RECORDS TO APPROPRIATE CUMSEG RECORDS, X-ARC  *
C     *               AND Y-ARC RECORDS FOR THE SAME YEAR,GRADE,AND *
C     *               SPECIALTY ARE COMBINED, AS WELL AS TREQ AND CREQ *
C     *               RECORDS OF THE SAME YEAR SPECIALTY AND GRADE  *
C     *       DATE: 15 APRIL 76                                     *
C     *       AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *       COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY           *
C     *               8120 WOODMONT AVENUE                          *
C     *               BETHESDA,MARYLAND 20014                       *
C     *                                                             *
C     *                                                             *
C     ***************************************************************
601   FORMAT(A2,J6,1X,3J12,I4,T9,I1)
602   FORMAT('0MESSAGE FROM SUBROUTINE RECORD'/
     *'0COMPUTATIONAL ADJUSTMENT MADE ON DATA BASE '/
     *'0REQUIREMENTS  RECORD - Z',J6/
     *'0ACTIVITY WAS ',I6,', UPPER LIMIT WAS ',I6/
     *'0THIS ADJUSTMENT NECESSARY DUE TO PROMOTEES '/
     *'0REPLACING THOSE ATTRITING DURING THE INTERVAL'/
     *'0ACTIVITY WAS SET EQUAL TO UPPER LIMIT')
        DIMENSION IN(3),ISUM(3)
      DIMENSION A(7),PARTBL(13)
        INTEGER A,ASAVE
      EQUIVALENCE (A(2),INDEX),(A(3),IN(1)),(A(6),IDATA),(A(7),ISEG)
      DATA PARTBL/1,1,12,0,0,1,
     * 2,1,36,0,0,2,99999/
        JINDEX=0
      CALL SOPEN3($100,$300,7,2,PARTBL)
C       FILE 7 IS THE CURSEG DATE BASE FILE
100   READ(7,601,END=299,ERR=299) A
C       CHANGE X AND Y COL RECORDS TYPE/PREFIX FOR SORTING
      IF(A(1).NE.'RN'.AND.A(1).NE.'CW'.AND.A(1).NE.'RV'
     *.AND.A(1).NE.'RU'.AND.A(1).NE.'RT')A(1)='ZZ'
C       CHANGE W COL RECORDS TYPE/PREFIX FOR SORTING
        IF(A(1).EQ.'CW')A(1)='WW'
C       RELEASE RECORD TO SORT PROGRAM
      CALL SRREL(A,7)
      GO TO 100
299   READ(18,601,END=200,ERR=200)A     @ FILE 18 IS THE OLD CUMULATIVE DATA
C       CHANGE X AND Y COL RECORDS TYPE/PREFIX FOR SORTING
      IF(A(1).NE.'RN'.AND.A(1).NE.'CW'.AND.A(1).NE.'RV'
     *.AND.A(1).NE.'RU'.AND.A(1).NE.'RT')A(1)='ZZ'
C       CHANGE W COL RECORDS TYPE/PREFIX FOR SORTING
        IF(A(1).EQ.'CW')A(1)='WW'
```

```
C          RELEASE RECORD TO SORT PROGRAM
      CALL SRREL(A,7)
      GO TO 299
200    CALL SSORT
300    REWIND 7
C          RELEASE SORT RECORDS
301    CALL SRRET(A,N,$500)
C          CHANGE TYPE/PREFIX BACK TO ORIGINAL
           IF( A(1).EQ.'WW') A(1) = 'CW'
           IF( A(1).EQ.'ZZ') A(1) = 'CZ'
C          IS THIS THE FIRST RECORD - YES,SAVE DATA
           IF(JINDEX .EQ. 0) GO TO 3
           IF (INDEX .NE. JINDEX) GO TO 2
C          IF YEAR,SPEC1,SPEC2 AND GRADE MATCH-ADD ACTIVITY,LB AND UB
           ISUM(3)=ISUM(2)+ISUM(1)
           ISUM(1) = IN(1)  + ISUM(1)
           ISUM(2) = IN(2)  + ISUM(2)
C          SAVE Y-ARC ACTIVITY IN LB FIELD OF Z RECORD
      IF(ISEQ.GT.2.AND.A(1).EQ.'CZ')ISUM(2)=ISUM(2)+IN(1)
           IF(A(1).NE.'RN') GO TO 400
           ISUM(2)=0
C          TEST IF COMBINED ACTIVITY GT UL
           IF(ISUM(1).LE.ISUM(3))GO TO 301
           WRITE(6,602)JINDEX,ISUM(1),ISUM(3)
           ISUM(1)=ISUM(3)
            GO TO 301
400        ISUM(3) = IN(3)  + ISUM(3)
           GO TO 301
C          WRITE CUMSEG RECORD
2          WRITE(3,904)ASAVE,JINDEX,ISUM,JDATA
904        FORMAT(A2,J8,'0',3J12,J4)
C          SAVE FIRST RECORD FOR COMPARISON W/SECOND
C          SAVE ACTIVITY,LL,UL RESPECTIVELY
3          ISUM(1) = IN(1)
           ISUM(2) = IN(2)
C          IF A Y-ARC RECORD,PUT Y-ACTIVITY IN LL FIELD
      IF(ISEQ.GT.2.AND.A(1).EQ.'CZ')ISUM(2)=IN(1)
           ISUM(3) = IN(3)
C          SAVE REST OF RECORD DATA
           JINDEX = INDEX
           ASAVE = A(1)
           JDATA = IDATA
           GO TO 301
C          WRITE LAST RECORD
500        WRITE(3,904)ASAVE,JINDEX,ISUM,JDATA
           ENDFILE 3
           ENDFILE 3
      RETURN
           END
```

```
C     ****************************************************************
C     ****************************************************************
C     *                                                              *
C     *          PROGRAM: UPDATE                                     *
C     *          INPUT FILES: ODSACUD01,                             *
C     *                       ODPOPUD01,                             *
C     *          OUTPUT FILES: ODSACUD01,                            *
C     *                        ODPOPUD01,                            *
C     *          PURPOSE: THIS PROGRAM UPDATES TOTAL REQUIREMENTS    *
C     *                   AND / OR RATES FOR GRADES O-6 THRU O-2.    *
C     *          DATE: 15 APRIL 78                                   *
C     *          AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *          COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY         *
C     *                   8120 WOODMONT AVENUE                       *
C     *                   BETHESDA,MARYLAND 20014                    *
C     *                                                              *
C     ****************************************************************
      DIMENSION RATE(7,31),IRATE(7,31),ITREQ(11,6,100),TEMP(9)
      INTEGER YEAR,GRADE,SPEC,OLDVAL
      EQUIVALENCE (RATE(1,1),IRATE(1,1))
   10 FORMAT ( )
   15 FORMAT (1H0,'CAPTAIN REMAINING RATES FOR GRADE',I2,1X,'ARE OUT OF
     1 SEQUENCE---INPUT IGNORED'/)
   20 FORMAT (1X,'REQUIREMENTS FOR GRADE',I2,2X,'SPECIALTY',I3,2X,'AT YE
     1AR',I3,' IS NOT EQUAL TO',I6,' IT IS EQUAL TO',I6,' , INPUT IGNORE
     1D',/)
   30 FORMAT (1X,'THE TOTAL REQUIREMENTS FOR GRADE',I2,2X,'SPECIALTY',I3
     1,2X,'AT YEAR',I3,' WAS',I6,' IT IS NOW',I6)
   40 FORMAT (1X,'GRADE',I3,1X,'WAS ENTERED ERRONEOUSLY')
   50 FORMAT (1X,'NEW',1X,A6,1X,'VALUES =',9(2X,F6.4)/)
   60 FORMAT (1X,'NEW',1X,A3,1X,'VALUES ARE',20(2X,I2)/)
   70 FORMAT (1H0,'GRADE O-',I1,1X,'CHANGES:')
   80 FORMAT (I1,A6)
   90 FORMAT (1H0,'NO. OF AES EXCEEDED 20----PROGRAM TERMINATED')
      IO=10
      NBRAES=0
C
C     **** READ IN WHETHER TO UPDATE REQUIREMENTS AND / OR RATES ****
C     **** 1=UPDATE REQUIREMENTS ****
C     **** 2=UPDATE RATES ****
C     **** 3=UPDATE REQUIREMENTS AND RATES ****
C
      READ (5,10) IPNTR
      GO TO (100,200,100),IPNTR
C
C     **** UPDATE TOTAL REQUIREMENTS ****
C
  100 READ (IO) ITREQ
C     READ IN YEAR,GRADE,SPECIALTY,OLD VALUE,AND NEW VALUE
  110 READ (5,10,END=130) YEAR,GRADE,SPEC,OLDVAL,NEWVAL
C     EDIT CHECK FOR MATCH ON OLD VALUE
C     IF EQUAL UPDATE REQ.,IF NOT EQUAL REJECT
      IF (ITREQ(YEAR,GRADE,SPEC) .EQ. OLDVAL) GO TO 120
      WRITE (6,20) GRADE,SPEC,YEAR,OLDVAL,ITREQ(YEAR,GRADE,SPEC)
      GO TO 110
```

```
  120 ITREQ(YEAR,GRADE,SPEC)=NEWVAL
      WRITE (6,30) GRADE,SPEC,YEAR,OLDVAL,NEWVAL
      GO TO 110
  130 REWIND 10
C     WRITE OUT UPDATED REQUIREMENTS
      WRITE (10) ITREQ
      IF (IPNYR .EQ. 1) STOP ONE
C
C     **** UPDATE RATES ****
C
  200 READ (11) RATE
C     READ IN GRADE AND RATE TYPE TO BE UPDATED
  210 READ (5,80,END=250) GRADE,R8TYPE
      IF (GRADE .GT. 1 .AND. GRADE .LE. 6) GO TO 220
      WRITE (6,40) GRADE
      GO TO 210
  220 IF (R8TYPE .EQ. 'ATTHI') READ (5,10) (RATE(GRADE,J),J=1,9)
      IF (R8TYPE .EQ. 'ATTLO') READ (5,10) (RATE(GRADE,J),J=10,18)
      IF (R8TYPE .EQ. 'PRMT') READ (5,10) (RATE(GRADE,J),J=19,27)
      IF (R8TYPE .EQ. 'OFLOHI') READ (5,10) RATE(GRADE,28)
      IF (R8TYPE .EQ. 'UFLOHI') READ (5,10) RATE(GRADE,29)
      IF (R8TYPE .EQ. 'OFLOLO') READ (5,10) RATE(GRADE,30)
      IF (R8TYPE .EQ. 'UFLOLO') READ (5,10) RATE(GRADE,31)
      IF (R8TYPE .NE. 'CPTREM') GO TO 240
C
C     READ IN CPTREM VALUES
      READ (5,10) (TEMP(J),J=1,9)
C     EDIT CHECK FOR SEQUENCE OF HI TO LO VALUES
      DO 230 J=1,8
      K=J+1
      L=J+9
      R8VAL=TEMP(J)*RATE(GRADE,L)
      IF (TEMP(K) .LE. R8VAL) GO TO 230
      WRITE (6,15) GRADE
      GO TO 240
  230 CONTINUE
C
C     SAVE CPTREM VALUES
      DO 235 J=1,9
      K=J+20
      IF (GRADE .EQ. 3) RATE(1,K)=TEMP(J)
      IF (GRADE .EQ. 2) RATE(7,J)=TEMP(J)
  235 CONTINUE
  240 IF (R8TYPE .EQ. 'AES') READ (5,10) NBRAES
      IF (NBRAES .GT. 20) WRITE (6,90)
      IF (NBRAES .GT. 20) STOP AES
      IF (R8TYPE .EQ. 'AES') READ (5,10) (IRATE(1,J),J=1,NBRAES)
      IF (R8TYPE .EQ. 'AES') IRATE(1,30)=NBRAES
C
C     WRITEOUT NEW RATE ARRAY VALUES
      WRITE (6,70) GRADE
      IF (R8TYPE .EQ. 'ATTHI') WRITE (6,50) R8TYPE,(RATE(GRADE,J),J=1,9)
      IF (R8TYPE .EQ. 'ATTLO')WRITE (6,50)R8TYPE,(RATE(GRADE,J),J=10,18)
      IF (R8TYPE .EQ. 'PRMT') WRITE (6,50)R8TYPE,(RATE(GRADE,J),J=19,27)
      IF (R8TYPE .EQ. 'OFLOHI') WRITE (6,50) R8TYPE,RATE(GRADE,28)
      IF (R8TYPE .EQ. 'UFLOHI') WRITE (6,50) R8TYPE,RATE(GRADE,29)
      IF (R8TYPE .EQ. 'OFLOLO') WRITE (6,50) R8TYPE,RATE(GRADE,30)
```

```
      IF (RBTYPE .EQ. 'UFLOLO') WRITE (6,50) RBTYPE,RATE(GRADE,3))
      IF (RBTYPE .EQ. 'CPTREM' .AND. GRADE .EQ. 3) WRITE (6,50) RBTYPE,
     1(RATE(1,J),J=21,29)
      IF (RBTYPE .EQ. 'CPTREM' .AND. GRADE .EQ. 2) WRITE (6,50) RBTYPE,
     1(RATE(7,J),J=1,9)
      IF (RBTYPE .EQ. 'AES') WRITE (6,60) RBTYPE,((RATE(1,J),J=1,NBRAES)
      GO TO 210
  250 REWIND 11
C     WRITE OUT UPDATED RATE VALUES
      WRITE (11) RATE
      STOP FINISH
      END
```

```
      SUBROUTINE KTREQ(ITREQ,JGRADE,NYRS,NUMBER,NSPEC,NBRSPC,ISEG)
C     ***********************************************************************
C     *                                                                     *
C     *         SUBROUTINE: KTREQ                                           *
C     *         CALLED BY: LINKAGE                                          *
C     *         CALLING ARGUMENTS:                                          *
C     *                             ITREQ- REQUIREMENTS                     *
C     *                             JGRADE- CURRENT GRADE                   *
C     *                             NYRS - NUMBER OF YEARS IN MODEL         *
C     *                             NUMBER - DUMMY                          *
C     *                             NSPEC - NUMBER OF SPECIALTIES           *
C     *                             NBRSPC - SPECIALTY NUMBERS              *
C     *                             ISEG - CURRENT SEGMENT NUMBER           *
C     *         PURPOSE: THIS  SUBROUTINE PRINTS OUT THE REQUIREMENTS       *
C     *                  VALUES FOR JGRADE AND JGRADE+1 FROM YEAR 0 TO      *
C     *                  NYEARS BEFORE UPDATING AND AFTER UPDATING.         *
C     *         DATE: 15 APRIL 78                                           *
C     *         AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN      *
C     *         COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY                 *
C     *                  8120 WOODMONT AVENUE                               *
C     *                  BETHESDA,MARYLAND 20014                            *
C     *                                                                     *
C     ***********************************************************************
      DIMENSION NBRSPC(50)
C*****    THIS PROGRAM WILL PRINT OUT THE VALUE OF ITREQ
C*****    FOR  JGRADE FROM YEAR 0 TO NYEARS
C*****
        DIMENSION ITREQ(11,6,100)
       DATA  BEFORE,AFTER/'BEFORE','AFTER '/
901     FORMAT(' SPECIALTY ',I2,' REQUIREMENTS',11I8)
902   FORMAT(1H0,10X,'REQUIREMENTS FOR GRADE ',I1,' *** ',A6,' ***
     1 DERIVING UNFILLED REQUIREMENTS')
903     FORMAT('                              ',9(5X,'T',I1,1X))
      KGRADE = JGRADE
      IF (JGRADE.NE.6)KGRADE=JGRADE+1
C       SUPPRESS AFTER DISPLAY FOR JGRADE+1,IF JGRADE NE 6
        IF(JGRADE.NE.6 .AND.NUMBER.EQ.2 .AND.ISEG.NE.1)KGRADE=JGRADE
        DO 30 I = JGRADE,KGRADE
      IF (NUMBER .EQ. 1) WRITE(6,902)I,BEFORE
      IF (NUMBER .EQ. 2) WRITE(6,902)I,AFTER
        WRITE(6,903)(L,L=0,8)
        DO 10    IK = 1,NSPEC
        K = NBRSPC(IK)
        DO 20  J = 1,11
 20     ITOTAL = ITOTAL + ITREQ(J,I,K)
C  IF NO REQUIREMENTS FOR A SPECIALTY IN ALL YEARS,SPECIALTY IS INVALID
        IF (ITOTAL .EQ. 0) GO TO 10
        WRITE(6,901) K,(ITREQ(J,I,K),J = 1,NYRS)
        ITOTAL = 0
10      CONTINUE
30      CONTINUE
        RETURN
        END
```

```
C      *****************************************************************
C      *                                                               *
C      *      PROGRAM: *LINKAGE                                        *
C      *      INPUT FILES: ODPOPUD01,                                  *
C      *                   ODSOLUD1,                                   *
C      *                   ODINPUD01,                                  *
C      *                   ODSACUD01,                                  *
C      *      OUTPUT FILES:                                            *
C      *                   ODINPUD01,                                  *
C      *                   ODSACUD01,                                  *
C      *                   11,    TEMP FILE                            *
C      *      CALLED ROUTINES: KTREQ,MODIFY(INTERNAL)                  *
C      *      PURPOSE: THIS PROGRAM UPDATES INPUT DATA FILE AND        *
C      *      MODIFIES THE TOTAL REQUIREMENTS FILES INORDER TO         *
C      *      RUN THE O-5 THRU O-2 GRADE SEGMENTS.                     *
C      *      DATE: 15 APRIL 76                                        *
C      *      AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN   *
C      *      COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY              *
C      *               8120 WOODMONT AVENUE                            *
C      *               BETHESDA,MARYLAND 20014                         *
C      *                                                               *
C      *****************************************************************
       COMMON /MODIFY/ ITREQ(11,6,100),RATE(7,31),LN,LYR,NSPECL,ULIMIT,
      1ACTIVT,JGRADE,NYRS,CORT,ISEG,SARRAY(50),NBRSPC(50)
C
       DATA NBRSPC/11,12,13,14,15,21,25,26,27,28,31,35,36,37,41,42,43,44,
      145,46,47,48,49,51,52,53,54,71,72,73,74,75,76,77,81,82,83,86,87,88,
      291,92,93,95,97,0,0,0,0,0/
C
       DOUBLE PRECISION SOLFIL(60),SOLFX(63)
       DIMENSION ISLFL(126),ISAVE(18),ITEMP(126),IRATE(7,31),UPBND(50)
       INTEGER SPECLT(50)
       DIMENSION AUTHMX(6)
       DATA IO,KCT/14,999/
       EQUIVALENCE (SOLFIL(1),ISLFL(1),SOLFX(1))
       EQUIVALENCE (RATE(1,1),IRATE(1,1))
C
C                           *****FORMAT STATEMENTS*****
C
15     FORMAT (9(I2,',',),I2)
20     FORMAT (8(F6.4,','),F6.4)
25     FORMAT (6(F10.0,','),1X)
30     FORMAT (I2,2X,I2,3X,2A4,2X,A4,I1)
40     FORMAT (1H1,'NO. OF RECORDS =',I5)
50     FORMAT ()
```

```
60      FORMAT (////)
70      FORMAT (A1)
80      FORMAT (10F6.0)
90      FORMAT (A1,A1,I2)
95      FORMAT (A1,I1,I2)
98          FORMAT(I2)
100     FORMAT (10F6.3)
110     FORMAT (1X,22I3)
120     FORMAT (1X,9F8.5)
C
        CALL OPT(IVALUE)
C
C       READ IN PARAMETERS FROM OLD INPUT DATA FILE
C       READ IN PROHIBITED ALTERNATE SPECIALTIES
C       READ IN ATTHI,ATTLO,PRMT, AND OVER/UNDER FLOW RATES
C
        READ (5,30) NSPEC,NYRS,NAME1,NAME2,MODE,JGRADE
        READ (5,50) AUTHMX
        READ (5,50) NPRO,(SPECLT(I),I=1,NPRO)
C       SKIP 4 RATE CARDS
        IF(JGRADE.LT.4) GO TO 125
        READ (5,60)
        READ (5,50) ISEG
        READ (5,80) (UPBND(K),K=1,50)
        READ (5,100) (SARRAY(K),K=1,50)
125     READ (12) RATE
C       READ IN TOTAL REQUIREMENTS
        READ (10) ITREQ
C       CALL EXTERNAL SUBROUTINE TO PRINT OUT ITREQ BEFORE UPDATING
        CALL KTREQ (ITREQ,JGRADE,NYRS+1,1,NSPEC,NBRSPC,ISEG)
C       READ IN A SOLUTION RECORD TO UPDATE ITREQ
C       FIRST TWO RECORDS ARE NOT USED
        READ (10) SOLFIL
        READ (10) SOLFIL
130     READ (10,END=390) SOLFIL
C       CHECK FOR CHANGE IN ROW/COLUMN RECORD
        IF (KCT.NE.999) GO TO 140
C       INITIALIZE POINTERS AND COUNTERS FOR ROW/COLUMN RECORDS
        KCT=0
        ICT=2
        IFOR2=4
        IFOR3=5
        IFOR4=6
140     ICT=ICT+1
        ICOUNT=ICT
C       CHECK FOR LAST ROW OR COLUMN RECORD
        IF (ISLFL(5).GE.0) GO TO 380
C       CHECK FOR TYPE 2 RECORD
150     IF (ICT.EQ.IFOR2) GO TO 210
C       CHECK FOR TYPE 3 RECORD
        IF (ICT.EQ.IFOR3) GO TO 270
C       CHECK FOR TYPE 4 RECORD
        IF (ICT.EQ.IFOR4) GO TO 330
C       **** TYPE 1 RECORD PROCESSING
        LAST=4
160     CONTINUE
C
```

```
C       SAVE LAST 18 WORDS FOR TYPE 2 RECORD PROCESSING
C
        DO 170 J=1,18
        K=J+102
        ISAVE(J)=ISLFL(K)
170     CONTINUE
C       EVALUATE DATA IN LAST RECORDS
180     IF (ILAST.NE.0) LAST=ILAST
        ILAST=0
        IBGN=7
        K1=IBGN+6
        K2=K1+3
        DO 200 J=1,LAST
        NK=(K2+2)/2
        IK=NK+6
C       GET ROW/COLUMN NAME FROM SOLUTION FILE
        IROW=ISLFL(K1)
C       GET UPPER LIMIT FROM SOLUTION  FILE
        ULIMIT=SOLFX(IK)
C       GET ACTIVITY FROM SOLUTION FILE
        ACTIVY=SOLFX(NK)
C       CHECK FOR 'XN' RECORDS
        DECODE (4,90,IROW) LN,LYR,NSPECL
        IF (LN.EQ.'X'.AND.LYR.EQ.'N') GO TO 190
        DECODE (4,95,IROW) LN,LYR,NSPECL
         IROWX=ISLFL(K1+1)
         DECODE(1,70,IROWX) CORT
         IF(JGRADE.EQ.2 .AND.LN.EQ.'X'.AND.LYR.EQ.0)
       .DECODE(2,98,IROWX)NSPECL
C       CALL SUBROUTINE TO MODIFY TOTAL REQUIREMENTS
190     CALL MODIFY
        CORT='N'
        K1=K1+24
        K2=K2+24
200     CONTINUE
        IF (KNT106.EQ.2) GO TO 390
        GO TO 130
C       **** TYPE 2 RECORD PROCESSING
210     DO 220 J=1,6
        ITEMP(J)=ISLFL(J)
220     CONTINUE
C
        DO 230 J=7,24
        K=J-6
        ITEMP(J)=ISAVE(K)
230     CONTINUE
C
        DO 240 J=7,108
        K=J+18
        ITEMP(K)=ISLFL(J)
240     CONTINUE
C
C       SAVE LAST 12 WORDS FOR TYPE 3 RECORD PROCESSING
C
        DO 250 J=1,12
        K=J+108
        ISAVE(J)=ISLFL(K)
```

```
250   CONTINUE
C
      DO 260 J=1,126
      ISLFL(J)=ITEMP(J)
260   CONTINUE
      IFOR2=IFOR2+4
      LAST=5
      GO TO 180
C     **** TYPE 3 RECORD PROCESSING
270   DO 280 J=1,6
      ITEMP(J)=ISLFL(J)
280   CONTINUE
C
      DO 290 J=7,18
      K=J-6
      ITEMP(J)=ISAVE(K)
290   CONTINUE
C
      DO 300 J=19,126
      K=J-12
      ITEMP(J)=ISLFL(K)
300   CONTINUE
C
C     SAVE LAST 6 WORDS FOR TYPE 4 RECORD PROCESSING
C
      DO 310 J=1,6
      K=J+114
      ISAVE(J)=ISLFL(K)
310   CONTINUE
C
      DO 320 J=1,126
      ISLFL(J)=ITEMP(J)
320   CONTINUE
      IFOR3=IFOR3+4
      LAST=5
      GO TO 180
C     **** TYPE 4 RECORD PROCESSING
330   DO 340 J=1,6
      ITEMP(J)=ISLFL(J)
340   CONTINUE
C
      DO 350 J=7,12
      K=J-6
      ITEMP(J)=ISAVE(K)
350   CONTINUE
C
      DO 360 J=13,126
      K=J-6
      ITEMP(J)=ISLFL(K)
360   CONTINUE
C
      DO 370 J=1,126
      ISLFL(J)=ITEMP(J)
370   CONTINUE
      IFOR4=IFOR4+4
      LAST=5
      GO TO 180
```

```
C      •••• LAST ROW OR COLUMN ADJUSTMENT TO INDEX-LAST
380    ILAST=(FLOAT(ISLFL(6))-3.0)/12.0+.5
       IF (ICT.EQ.IFOR2) ILAST=ILAST+1
       WRITE(6,11)ILAST,ICOUNT
11     FORMAT('0 LAST WORD IS',I5,' IN RECORD NBR',I5)
       KNTI06=KNTI06+1
       KCT=999
       IF (KNTI06.EQ.1) GO TO 150
       IF (ICT.EQ.IFOR2) GO TO 210
       IF (ICT.EQ.IFOR3) GO TO 270
       IF (ICT.EQ.IFOR4) GO TO 330
       GO TO 160
390    WRITE (6,40) ICOUNT
C      SAVE UPDATED TOTAL REQUIREMENTS MATRIX
       REWIND 10
       WRITE (10) ITREQ
C      CALL EXTERNAL SUBROUTINE TO PRINT OUT ITREQ AFTER UPDATING
       CALL KTREQ (ITREQ,JGRADE,NYRS+1,2,NSPEC,NBRSPC,ISEG)
C      UPDATE INPUT DATA FILE
C          SAVE ORIGINAL ISEG VALUE
        ISEGX=ISEG
C      WRITE PARAMETERS
       IF (ISEG.NE.1) JGRADE=JGRADE-1
C          •••DO NOT WRITE TO FILE 11 IF GRADE = 1 •••
           IF(JGRADE.EQ.1)GO TO 435
       WRITE (11,30) NSPEC,NYRS,NAME1,NAME2,MODE,JGRADE
       WRITE (11,25) AUTHMX
C      WRITE PROHIBITED ALTERNATE SPECIALTIES
       WRITE (11,15) NPRO,(SPECLT(I),I=1,NPRO)
C      WRITE HIGH ATTRITION RATES
       WRITE (11,20) (RATE(JGRADE,K),K=1,9)
C      WRITE LOW ATTRITION RATES
       WRITE (11,20) (RATE(JGRADE,K),K=10,18)
C      WRITE PROMOTION RATES
       WRITE (11,20) (RATE(JGRADE,K),K=19,27)
C      WRITE OVER AND UNDER FLOW VARIABLES
       WRITE (11,20) (RATE(JGRADE,K),K=28,31)
       IF (JGRADE.LT.4) GO TO 420
       IF (ISEG.NE.1) GO TO 400
       ISEG=ISEG+1
       WRITE (11,50) ISEG
       WRITE (11,80) (UPBND(K),K=1,50)
       DO 395 K=1,50
       SARRAY(K)=1.0
395    CONTINUE
       WRITE (11,100) (SARRAY(K),K=1,50)
400    IREAD=(JGRADE-4)
        IF(ISEGX .NE. 1)IREAD=IREAD +1
       IF(IREAD .LT.1)END FILE 11
       IF(IREAD .LT.1)STOP
       DO 410 K=1,IREAD
       READ (5,50) ISEG
       WRITE (11,50) ISEG
       READ (5,80) (UPBND(J),J=1,50)
       WRITE (11,80) (UPBND(J),J=1,50)
       READ (5,100) (SARRAY(J),J=1,50)
       WRITE (11,100) (SARRAY(J),J=1,50)
```

```
410     CONTINUE
C       CHECK FOR GRADE 3
        IF (JGRADE.GT.3) STOP
420     IF (JGRADE.EQ.2) GO TO 430
C       WRITE OUT CAPTAIN PARAMETERS
C       WRITE OUT 20 AES,ICHG,NBRAES
        WRITE (11,110) (IRATE(1,K),K=1,20),IRATE(1,31),IRATE(1,30)
C       WRITE OUT CAPTAINS REMAINING BY YEAR
        WRITE (11,120) (RATE(1,K),K=21,29)
        STOP
C       WRITE OUT LIEUTENANTS PARAMETERS
C       WRITE OUT 20 AES,ICHG,NBRAES
430     WRITE (11,110) (IRATE(1,K),K=1,20),IRATE(7,10),IRATE(1,30)
C       WRITE OUT CAPTAINS REMAINING BY YEAR
        WRITE (11,120) (RATE(7,K),K=1,9)
435        END FILE 11
        STOP
C
C       INTERNAL SUBROUTINE MODIFY
C       THIS SUBROUTINE UPDATES THE TOTAL REQUIREMENTS FILE
C
        SUBROUTINE MODIFY
        COMMON /MODIFY/ ITREQ(11,6,100),RATE(7,31),LN,LYR,NSPECL,ULIMIT,
       1ACTIVY,JGRADE,NYRS,CORT,ISEG,SARRAY(50),NBRSPC(50)
C
C
C                                  *****FORMAT STATEMENTS*****
C
440     FORMAT ('0LYR  KYR SPECL JGRADE INDEX REQONE REQTWO REQTOT ',
       1' TOTREQ RATEX RATEY SARRAY  ULIM   CRQACT',/,I4,2I5,2I6,
       22X,3F7.1,F8.1,F4.1,3X,F4.1,2X,F5.1,2F7.1)
450     FORMAT (' ITREQ = ',I4,' XI=',F8.2,' ACT=',F8.2,' RATE=',F8.2,
       1' ULIMIT = ',F10.2,' CORT = ',A1)
458     FORMAT (' ITREQ = ',I4,' ACT=',F8.2)
459        FORMAT(1H0,'INVALID SPECIALTY ENCOUNTERED IN MODIFY ')
C
C          BYPASS CREQ RECORD FOR ISEG EQUAL 0 OR 2
          IF(CORT.EQ. 'C' .AND. ISEG.NE. 1) RETURN
        J=1
C       **SET YEAR FOR TREQ AND CREQ RECORDS
        IF (LN.EQ.'N') KYR=LYR+1
C
C       **SET YEAR FOR XN--VARIABLE DATA
        IF (LN.EQ.'X'.AND.LYR.NE.0) KYR=NYRS+1
C       **SET YEAR FOR X000-- VARIABLE DATA(LT SEGMENT ONLY)
            IF(LN.EQ.'X'.AND.LYR.EQ.0)KYR=1
        IF (LN.EQ.'X') J=2
        INDEX=JGRADE
        IF (JGRADE.LT.6) INDEX=JGRADE+1
        REQONE=ITREQ(KYR,INDEX,NSPECL)
C       **VALIDATE SPECIALTY NUMBER
          DO 455 K=1,50
          IF(NSPECL.NE. NBRSPC(K))GOTO455
          IN=K
          GO TO 456
455       CONTINUE
          WRITE (6,459)
```

```
      RETURN 0
456       ULIM=ULIMIT
          IF(JGRADE.GT.3)ULIM=ULIMIT/SARRAY(IN)
      REQTWO=MAX((ULIM-REQONE*(1.+RATE(INDEX,30)))/(1.+RATE(JGRADE,28))
     1,0.)
      REQTOT=(REQONE+REQTWO)*SARRAY(IN)
C         SARRAY(1)=0 FOR LINK TO LT SEGMENT
          IF(JGRADE.LT.4)REQTOT=REQONE + REQTWO
      TOTREQ=REQTOT-ACTIVY
      RATEX=RATE(INDEX,30)
      RATEY=RATE(JGRADE,28)
C     **IF OPTION ON XQT CARD,THEN PRINT KEY VARIABLES
      IF(IVALUE.GT.0)
     *WRITE (6,440) LYR,KYR,NSPECL,JGRADE,INDEX,REQONE,REQTWO,REQTOT,
     1TOTREQ,RATEX,RATEY,SARRAY(IN),ULIM,CRQACT
C     **TEST FOR SEGMENT 0 OR 2,6 ROW OR COLUMN RECORD
      IF (ISEG.NE.1) GO TO (460,480), J
C     IN PHASE I
C     **SEGMENT ONE TEST FOR ROW OR COLUMN RECORD
      GO TO (500,480), J
C
C     *** N---TREQ *** CONSTRAINT DATA FOUND
C
460   IPNTR=KYR+9
C         *XI IS ATTRITION FOR LOWER GRADE IF IPNTR=KYR+9,
C         OTHERWISE FOR HIGHER GRADE*
      XI=ACTIVY*RATE(JGRADE,IPNTR)
      IF (ACTIVY.LT.ULIMIT) GO TO 470
      ITREQ(KYR,JGRADE,NSPECL)=MAX((XI+.5),1.)
      IF (KYR.EQ.1) ITREQ(KYR,JGRADE,NSPECL)=0
      GO TO 490
C
470       ITREQ(KYR,JGRADE,NSPECL)=MAX(1.,(ULIM-ACTIVY+XI)/(1.0+RATEX))
     *+.5
        IF(KYR.EQ.1)ITREQ(KYR,JGRADE,NSPECL)=(ULIM-ACTIVY)/(1.0+RATEX)
     *+.5
      GO TO 490
C
C     *** XN-- *** FOUND
C         ***ALSO X000-- IN LT SEGMENT***
480       ITREQ(KYR,JGRADE,NSPECL)=MAX(1.,(ULIM-ACTIVY)/(1.0+RATEX))+.5
          WRITE(6,458)ITREQ(KYR,JGRADE,NSPECL),ACTIVY
490   CONTINUE
C     **IF OPTION ON XQT CARD,THEN PRINT KEY VARIABLES
      IF (IVALUE.LT.1) RETURN
      IF(CORT.EQ.'T')
     *WRITE (6,450) ITREQ(KYR,JGRADE,NSPECL),XI,ACTIVY,
     1RATE(JGRADE,IPNTR), ULIMIT ,CORT
      IF(CORT.EQ.'C')
     *WRITE (6,450) ITREQ(KYR,INDEX,NSPECL),XI,ACTIVY,
     1RATE(INDEX,IPNTR), ULIMIT ,CORT
      RETURN
C
C     ***CREQ OR YREQ IN PHASE I***
500   IPNTR=KYR+9
C         *XI IS ATTRITION FOR LOWER GRADE IF IPNTR=KYR+9,
C         OTHERWISE FOR HIGHER GRADE*
      XI=ACTIVY*RATE(JGRADE,IPNTR)
      IF (CORT.EQ.'C') XI=ACTIVY*RATE(INDEX,KYR)
```

```
      IF (ACTIVY.LT.ULIMIT) GO TO 510
C     ACTIVITY=ULIMIT
       IF(CORT.EQ.'C') CRQACT=ACTIVY
      IF (CORT.EQ.'C') ITREQ(KYR,INDEX,NSPECL)=MAX((ITREQ(KYR,INDEX,
     1NSPECL)-ULIMIT+X1),1.)
         IF((KYR .GT. 1) .AND.
     A (CORT.EQ.'T'))ITREQ(KYR,JGRADE,NSPECL)=MAX((ITREQ(KYR,JGRADE,
     1NSPECL)-ULIMIT+X1),1.)
       IF (KYR.EQ.1) ITREQ(KYR,JGRADE,NSPECL)=
     *     MAX(((ITREQ(KYR,JGRADE,NSPECL)-ULIMIT)+.5),0)
      GO TO 490
C
C     ACTIVITY IS LESS THAN UPPER LIMIT FOR PHASE ONE
510   IF (CORT.NE.'C') GO TO 520
C     CREQ RECORD
      CRQACT=ACTIVY
      ITREQ(KYR,INDEX,NSPECL)=MAX(REQONE-ACTIVY+X1+.5,1.)
      GO TO 490
C
C        .ACTIVITY .LT. UPPER LIMIT-TREQ RECORD - PHASE 1.
520      IF(JGRADE.EQ.6)REQTWO=REQONE
       ITREQ(KYR,JGRADE,NSPECL)=MAX((REQTWO+CRQACT-ACTIVY+X1),1.)
       IF(KYR.EQ.1)ITREQ(KYR,JGRADE,NSPECL)=MAX((REQTWO+CRQACT
     * -ACTIVY),0.)
       GO TO 490
      END
```

```
C
C     ***********************************************************
C     *                                                         *
C     *     PROGRAM: SEPARATE                                   *
C     *     CALLED ROUTINES: NONE                               *
C     *     INPUT FILES:                                        *
C     *                    -18 - ODSAPUD18.                     *
C     *     OUTPUT FILES:                                       *
C     *                    -16 - (TEMP) COPIED TO ODSAPUD18.    *
C     *                    -17 - (TEMP)RECORDS NOT NEEDED IN ODSAPUD18. *
C     *     PURPOSE: THIS PROGRAM SEPARATES THE CURRENT CUMULATIVE *
C     *              DATA BASE INTO TWO COMPONENTS.              *
C     *              1) THOSE TO BE UPDATED IN FURTHER SEGMENTS  *
C     *              2) THAT OUTPUT READY TO GO INTO THE MIRADS  *
C     *                 DATA BASE W/O CHG                        *
C     *     DATE: 15 APRIL 76                                    *
C     *     AUTHORS: MAJ J.D. THOMAS,MAJ J.W. OLSON,MR. R.L. BROWN *
C     *     COMMAND: U.S. ARMY CONCEPTS ANALYSIS AGENCY          *
C     *              8120 WOODMONT AVENUE                        *
C     *              BETHESDA,MARYLAND 20014                     *
C     *                                                         *
C     ***********************************************************
      DIMENSION A(7)
      INTEGER A
      READ(5,900)JGRADE
900   FORMAT(I1)
1     READ(18,901,END=999)A
      IF(A(2).EQ.'97100')PRINT 903,A
903   FORMAT(1X,A2,A5,I1,1X,3I12,I4)
901   FORMAT(A2,A5,I1,1X,3I12,I4)
902   FORMAT(A2,A5,I1,'0',3J12,J4)
      IF(A(3).LE.(JGRADE+1))GO TO 2
      WRITE(17,902)A
      GO TO 1
2     WRITE(16,902)A
      GO TO 1
999   ENDFILE 17
      ENDFILE 17
      ENDFILE 16
      ENDFILE 16
      STOP
      END
```

APPENDIX A

CONTRIBUTORS

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

APPENDIX A
CONTRIBUTORS

1. <u>Documentalist</u>

   MAJ Joseph D. Thomas
   Methodology, Resources and Computation Directorate
   United States Army Concepts Analysis Agency (CAA)

2. <u>Technical Support Personnel</u>

   Mr. George E. Armstrong

3. <u>Editorial Support</u>

   Mr. Andrew J. Gamble

4. <u>Administrative Support Personnel</u>

   Mrs. Elizabeth Demshock
   Mrs. Thelma Laufer

APPENDIX B

REFERENCES

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

APPENDIX B
REFERENCES

1. Sperry UNIVAC 1100 Series, "Functional Mathematical Programing System (FMPS)," Bonner and Moore Associates, Inc., Dallas, Texas, 1975.

2. National Aeronautics and Space Administration, George C. Marshall Space Flight Center, "Marshall Information Retrieval and Display System, MIRADS-2, Implementation Manual," Montgomery, AL, June 1975.

3. National Aeronautics and Space Administration, George C. Marshall Space Flight Center, "Marshall Information Retrieval and Display System, MIRADS-2 User's Manual," Montgomery, AL, June 1975.

4. US Army Concepts Analysis Agency, "Officer Dual Specialty Allocation System (ODSAS)," Bethesda, MD, Study Report CAA-SR-76-6, April 1976.

APPENDIX C

GLOSSARY

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

APPENDIX C
GLOSSARY

1. Abbreviations, Acronyms, and Short Terms

ADP	automatic data processing
AES	advanced entry specialty
AFCS	active federal commissioned service
BES	basic entry specialty
CAA	Concepts Analysis Agency
COL	colonel
CONUS	Continental United States
CPT	captain
CRT	cathode ray tube (display terminal)
DA	Department of the Army
DCSPER	Deputy Chief of Staff for Personnel
DOPMS	Defense Officer Personnel Management System
EEA	essential elements of analysis
FMPS	Functional Mathematical Programing System
HUMINT	the intelligence collection function that uses human beings as both sources and collectors
HQ	headquarters
LP	linear programing
LT	lieutenant

LTC	lieutenant colonel
MAJ	major
MILPERCEN	United States Army Military Personnel Center
MTOE	Modification Table of Organization and Equipment
NASA	National Aeronautics and Space Administration
ODCSOPS	Office, Deputy Chief of Staff for Operations and Plans
ODCSPER	Office, Deputy Chief of Staff for Personnel
OPD	Officer Personnel Directorate
OPMD	Officer Personnel Management Directorate
OPMS	Officer Personnel Management System
ORSA	Operations Research/Systems Analysis
PERSACS	Personnel Structure and Composition System
pct auth	percentage authorized
RCS	Reports Control Symbol
SACS	structure and composition system
SACSTAPE1(2)	magnetic tape(s) containing SACS data
SAG	Study Advisory Group
TDA	tables of distribution and allowances
TOE	table(s) of organization and equipment
UIC	Unit Identification Code

USAMSSA	US Army Management Systems Support Agency
YOS	year(s) of service
1LT	first lieutenant
2LT	second lieutenant

2. Computer Models, Routines, Simulations, Related Terms, and Definitions

AID-O	Automated Interaction Detector-Officers. A model providing data on attrition rates and populations by grade and years of service.
BCD	Binary Coded Decimal Notation
BPI	Bits per inch
CIM-O	Central Integrating Model-Officers. A model that provides data on attrition rates and populations by grade and years of service.
DATA processor	Data processor of EXEC 8 operating system
EBCDIC	Extended Binary Coded Decimal Interchange Code
ED processor	Edit processor of EXEC 8 operating system
ELT processor	Element processor of EXEC 8 operating system
EXEC 8	UNIVAC 1108 operating system
FIELDATA	A 6-bit character code that is the native character of the UNIVAC 1100-series computer system
FORTRAN	Formula Translation (a computer coding language used in scientific applications)

MIRADS	Marshall Information Retrieval and Display System
MPS-X	Mathematical Programing System-Extended
ODSAS	Officer Dual Specialty Allocation System
SPRINT	Specialized routine within FMPS that accelerates solution time

3. Terms Unique to ODSAS

a. General

BGNYOS	beginning year of service
CINC	unique suffix qualifier for name of one type of flow control for Y arcs
CREQ	unique suffix qualifier for name of capacity constraint on Y arc input to a node
ENDYOS	ending year of service
GOZO	unique suffix qualifier for name of flow conservation constraint
LINC	unique suffix qualifier for name of one type of flow control for X arcs
NPREF	total number of preferred, or logical, specialty pairings considered in arriving at a solution
NSPEC	total number of authorized OPMS specialties
NYRS	number of years in projection period
OBJECTIV	objective function name
PROM	promotion
Spec	specialty
SPEC-PAIRS	a standard MIRADS query set used to display specialty pairings

TOTAUTH	unique name for one type of control of input constraint, indicating total number of officers authorized for a grade segment at start date
TREQ	unique suffix qualifier for name of capacity constraint (X+Y input)
UBSG	unique suffix qualifier for name of control of input constraint for specialties

b. <u>ODSAS Catalogued Runstreams</u>. - See Chapter IX.

c. <u>ODSAS Disc and Tape File Descriptions</u>. - See Chapter VIII.

d. <u>ODSAS Programs (Major Routines and Subroutines)</u>. - See Chapters XI and XII.

e. <u>ODSAS Programs (Minor Supporting Functions and Subroutines)</u>. - See Table III-1, Chapter III, and Chapter XII.

f. <u>ODSAS Variable Names</u>. - See Chapter X.

4. <u>MIRADS-Created Data Base Records</u>. - See Chapter VII.

APPENDIX D

ODSAS NAMING CONVENTIONS

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

APPENDIX D
ODSAS NAMING CONVENTIONS

1. General. - Three naming conventions for the LP variables and constraints are utilized in ODSAS to facilitate interpretation and analysis. One convention provides for unique identification of variables with the arcs with which they are associated. The second convention assigns unique names to the rows of the LP matrix output by the FMPS activity during the processing phase of the ODSAS automated information system. The third convention is a modified version of the first, and is used only in connection with the columns portion of the FMPS output. These three conventions are described respectively in Paragraphs 2, 3 and 4, below.

2. Naming Convention for Variables. - This convention was devised so that the variables in the constraints used in each grade segment could be uniquely identified with network arcs and have an intrinsic meaning. The naming convention for the variables consists of four fields, one alphabetic followed by three numeric subscripts, i.e., the general form is A_{yfg}. Each term is explained as follows:

A	y	f	g
Alphabetic identifier	Year coming from	Specialty number coming from	Specialty number going to

Where,

A = W - identifier for arcs in the $T'_{\emptyset}$ to $T_{\emptyset}$ interval only

X - identifier for officers remaining in grade

Y - identifier for officers promoted to higher grade

y = Ø - 9

f = Ø1 - 99*

g = Ø1 - 99*

---

*Includes only the 45 currently authorized OPMS specialty numbers.

For example, an arc connecting the Specialty 25 node at $T_{\emptyset}$, and the Specialty 36 node at T_1, for a LTC in the LTC segment, would be $X_{\emptyset 2536}$; for a LTC promoted to COL in the same segment and year, and the same "from" and "to" nodes as the previous example, the arc would be $Y_{\emptyset 2536}$.

3. Naming Convention for FMPS Solution Output - Rows Section. - Rows are assigned unique names relating to constraint types (the five types of constraints are: (1) flow conservation, (2) node capacities, (3) control of flows for dual specialties, (4) control of input to the network, and (5) key arc relationships). Row names appear in one of the following five formats.

a. Format 1. - This format consists of one alphabetic character for an identifier (N or W), one number for year, two numbers identifying specialty, and a four-character alphabetic name. For example, NØ11TREQ is a constraint for year $T_{\emptyset}$, Specialty 11, for the Total REQuirements capacity. There are six possible four-character names: (1) GOZO, indicating flow conservation, (2) CINC, indicating flow control for Y arcs where "from" and "to" specialty numbers are identical, (3) LINC, indicating for X arcs the same as CINC indicates for Y arcs, (4) UBSG, indicating control of input for selected specialties, (5) CREQ, and (6) TREQ. (Both (5) and (6) are node capacity constraints.) Examples are shown in Figure V-12.

b. Format 2. - This format is one alphabetic character "R" and a five-character numeric identifier of an X arc. For example, RØ1121 is a flow control constraint upon arc $X_{\emptyset 1121}$. The R-named rows restrict the flow in X arcs. Examples are contained in Table D-1.

c. Format 3. - This format consists of three alphabetic characters "RES" and a five-character numeric identifier of a Y arc. For example, RESØ1121 is a flow control constraint upon arc $Y_{\emptyset 1121}$. The RES-named rows restrict the flow in Y arcs.

d. Format 4. - Two alphabetic characters, "UR," and a four-character numeric identifier of a predefined specialty pair; e.g., UR1121 is a key arc relationship constraint, constructed using the utilization ratio for Specialties 11 and 21, which relates the number of officers with 11/21 specialty pair serving in Specialty 11 to the number serving in Specialty 21. (Examples are contained in Figure V-12.)

e. Format 5. - This format consists of one row name "TOTAUTH," which identifies uniquely the control of input constraint for the total number of officers authorized at $T_{\emptyset}$ in the grade segment being processed. (Other control of input constraints are named according to Format 1, with year equaling Ø, the appropriate specialty number, and a four character name "UBSG" (meaning an upper limit in segment 1.))

A tabular recapitulation of these five row-naming convention formats is presented in Table D-1.

4. Special Case Naming Convention for FMPS Solution Output - Columns Section. - Within the Columns Section of the FMPS solution output, the naming convention described in Paragraph 2 above applies in most instances when identifying variables with their related arcs. However, there is one unique condition for which a modified convention is applied--that unique condition being when the variables represent arcs exiting the network. In such cases, the variables are assigned names in the form XN_{nn}, where N indicates the final year and nn is the specialty or node number (see example at bottom of Figure V-13). Names in this form appear only in those cases where the input cost (the coefficient of a variable in the objective function) has a value other than zero.

TABLE D-1, Correlation of Naming Convention Formats and Constraint Types

CONSTRAINT TYPE	FORMAT 1	FORMATS 2, 3, 4 & 5
	a (N or W) n (yr) nn (spec nr) aaaa (name)	
Flow conservation	Example: NØ11GOZO WØ11GOZO GOZO - everything that goes into a node must go out. N is normal identifier. W is substituted for N when the year is $T_{\emptyset}$. WnnnGOZO may appear in any segment except LT	N/A
Node capacities	Example: NØ11TREQ NØ11CREQ TREQ - total capacity (for the current grade plus the unfilled higher grade) CREQ - capacity for the unfilled higher grade requirements only	N/A
Flow control	Example: NØ11CINC CINC - control of Y arc when "from" and "to" specialties are identical Example: NØ11LINC LINC - control of X arc when "from" and "to" specialties are identical Example: WØ11LINC WnnnLINC is used in CPT segment only, to control $W_{\emptyset nnnn}$ arcs where "from" and "to" specialties are identical	FORMAT 2 a (R) nnnnn (X arc) Example: RØ1121 R-named rows restrict the flow in X arcs FORMAT 3 aaa (RES) nnnnn (Y arc) Example: RESØ1121 RES-named rows restrict the flow in Y arcs
Key arc relationship	N/A	FORMAT 4 aa (UR) nnnn (spec pair) Example: UR1121 UR-named rows are constraints upon the location of officers with the two specialties of the pair
Control of input	Example: WØ11UBSG UBSG - upper limit on number entering with specialty nn as a primary or alternate of grade being processed (applies to field grade segments only)	FORMAT 5 aaaaaaa (TOTAUTH) Example: TOTAUTH Total number of officers authorized in this grade segment at $T_{\emptyset}$

APPENDIX E

DISTRIBUTION

OFFICER DUAL SPECIALTY ALLOCATION SYSTEM
(ODSAS)

APPENDIX E
DISTRIBUTION

Addressee	Copies
1. Army Centers	
Commander US Army Military Personnel Center ATTN: DAPC-PSS-A 200 Stovall Street Alexandria, VA 22332	3
Commander US Army Military Personnel Center ATTN: DAPC-OPP 200 Stovall Street Alexandria, VA 22332	5
2. DoD Center	
Defense Documentation Center for Scientific and Technical Information Building 5 Cameron Station Alexandria, VA 22314	3
3. HQ DA	
Assistant Secretary of the Army (Manpower and Reserve Affairs) Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans Headquarters Department of the Army Washington, DC 20310	1
Deputy Under Secretary of the Army (Operations Research) Washington, DC 20310	1
Office of the Deputy Chief of Staff for Personnel Headquarters Department of the Army ATTN: DAPE-MPO Washington, DC 20310	5

Addressee	Copies
4. US Army Concepts Analysis Agency	
Chief, PP	1
Chief, AD, ATTN: Library Branch	2
Director, MR	3